



Universidad
Carlos III de Madrid
www.uc3m.es

Departamento de Teoría de la Señal y Comunicaciones
Ingeniería Técnica de Telecomunicación:
Sistemas de Telecomunicación

Proyecto Fin de Carrera

SCRUM

Aplicación del método ágil en la gestión de proyectos

Ana Retamosa Santos

Tutora: Raquel Pérez Leal

5 de Octubre de 2015

Av. Universidad, 30
28911 Leganés (Madrid)



Título: Scrum, Aplicación del método ágil en la gestión de proyectos.

Autora: Ana Retamosa Santos

Tutora: Raquel Pérez Leal

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa del Proyecto Fin de Carrera el día ____ de _____ de _____ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

Fdo. Presidente

Fdo. Secretario

Fdo. Vocal

AGRADECIMIENTOS

*A mi marido, Álvaro, que ha estado a mi lado cada día
dándome ese impulso que necesitaba para poder finalizar este proyecto.
Gracias por tu amor, por confiar en mí y ayudarme a creer que puedo hacerlo.*

*A mis padres, Antonio y Maribel, que gracias a ellos
he podido llegar donde estoy hoy y me han dado siempre su apoyo.
Gracias por vuestro cariño y por estar siempre a mi lado.*

*A mi hermana, Azahara, que ha sido siempre mi gran ejemplo.
Gracias por quererme tanto, escucharme y darme buenos consejos.*

*A mi cuñado, Julián, que me acompañó en los últimos años de universidad
y a mi sobrina, Alba, que siempre ha sabido sacarme una sonrisa.*

*A mis suegros, Conchi y Manolo, y a mi cuñada, Laura,
que siempre están dispuestos a ayudar.
Gracias por vuestro afecto y ternura.*

*A mi tutora, Raquel, que me ha ayudado
y guiado en el desarrollo de este proyecto.*

*Y por último, quiero dedicar este proyecto a mi hija Carla,
que con solo 8 meses es lo más grande que tengo en mi vida.*

RESUMEN

Las metodologías ágiles han aparecido en los últimos años como alternativa a las metodologías convencionales con carácter normativo y fuerte dependencia de planificaciones previas al desarrollo. Una metodología ágil es capaz de adaptarse a los cambios, especialmente de los requisitos, según el *feedback* constante y temprano del cliente.

En este proyecto se hace una breve exposición de la evolución de las metodologías hasta la llegada de las metodologías ágiles con la creación del Manifiesto Ágil, describiendo las más punteras y utilizadas a día de hoy enmarcándolas en un cuadro comparativo.

La esencia del proyecto se encuentra en el estudio de la metodología Scrum, ayudando al lector a entender su proceso con la definición de sus roles y los elementos que la conforman. Se pondrá en práctica dicha metodología en un proyecto real detallando la ejecución del mismo de una forma global.

El objetivo es facilitar una herramienta a aquellos que vienen utilizando las metodologías convencionales para introducirse en las metodologías ágiles y, al mismo tiempo, ganar confianza en esta forma novedosa de gestionar proyectos.

Palabras clave: Metodología, Ágil, Scrum, *Sprint*, Iteración, *Product Owner*, *Scrum Master*, Equipo, Gestión de proyectos

ABSTRACT

In recent years Agile methodologies have appeared as an alternative to conventional methods which have normative character and hard reliance to planning of pre-development. An agile methodology is able to adapt to changes, particularly to the requirements of the customer with whom you have a constant relationship.

This project contains a brief exposition of the evolution of the methodologies to the arrival of agile methodologies with the creation of the Agile Manifesto. In it described a summary of the most advanced and used methods today.

The essence of the project is in the study of the Scrum methodology, helping the reader to understand their process by defining their roles and the elements that comprise it. This methodology will be implemented in a real project detailing the implementation of all them.

The goal is to provide a tool for who are using conventional methods to enter agile methodologies and at the same time, to win confidence in this new way of managing projects.

Keywords: Methodology, Agile, Scrum, *Sprint*, Iteration, *Product Owner*, *Scrum Master*, Team, Project Management.

INDICE GENERAL

INDICE DE TABLAS.....	1
INDICE DE FIGURAS.....	2
1. INTRODUCCIÓN.....	4
1.1. Introducción.....	4
1.2. Objetivo	5
1.3. Motivación personal	6
1.4. Estructura de la memoria	6
2. METODOLOGÍAS ÁGILES DE GESTIÓN DE PROYECTOS	8
2.1. Historia de los procesos de desarrollo	8
2.2. Metodologías Ágiles	11
2.3. ¿Cuándo se aplica una Metodología Ágil?.....	14
2.4. Manifiesto Ágil.....	15
3. SCRUM	18
3.1. Introducción.....	18
3.2. Roles de Scrum	20
3.2.1. <i>Product Owner</i> (Propietario del Producto)	21
3.2.2. Equipo de Desarrollo (<i>Development Team</i>)	21
3.2.3. <i>Scrum Master</i> (Facilitador).....	22
3.3. Componentes (<i>Artefacts</i>)	23
3.3.1. Lista de Producto (<i>Product Backlog</i>)	23
3.3.2. Lista de Pendientes del <i>Sprint</i> (<i>Sprint Backlog</i>).....	24
3.3.3. Objetivo del <i>Sprint</i> (<i>Sprint Goal</i>)	25
3.3.4. Incremento	25
3.3.5. Terminado	25
3.4. Eventos	26
3.4.1. <i>Sprint</i>	26
3.4.2. Reunión de Planificación del <i>Sprint</i> (<i>Sprint Planning Meeting</i>)	27

3.4.3. Scrums Diarios (<i>Daily Scrums</i>)	28
3.4.4. Revisión del <i>Sprint</i> (<i>Sprint Review</i>).....	28
3.4.5. Retrospectiva del <i>Sprint</i> (<i>Sprint Retrospective</i>)	29
3.5. Otras Herramientas	30
3.5.1. <i>Planning Poker</i>	30
3.5.2. Pizarra de Tareas (<i>Scrum Taskboard</i>).....	31
3.5.3. Gráfica <i>Burn Down</i>	32
3.6. El proceso de Scrum	35
4. CASO DE ESTUDIO	38
4.1. Modelo Operativo	39
4.1.1. Alcance y Líneas de Servicio	39
4.1.1.1. Adaptación Nueva Convocatoria	40
4.1.1.2. Mantenimiento Correctivo	41
4.1.1.3. Mantenimiento Preventivo.....	41
4.1.1.4. Mantenimiento Evolutivo	42
4.1.1.5. Nuevos Desarrollos	42
4.1.1.6. Soporte a Usuarios.....	43
4.1.2. Definición de Proyecto y ciclo de vida	43
4.1.2.1. Fases Directivas.....	45
4.1.2.2. Fases Productivas.....	54
4.1.3. Seguimiento y Control.....	63
4.1.4. Proceso de Certificación de los trabajos realizados.....	64
4.2. Modelo Organizativo	66
4.2.1. Configuración Equipo de Trabajo	67
4.2.2. Roles y Responsabilidades	67
4.2.2.1. Roles del Cliente	67
4.2.2.2. Roles del Proveedor	70
4.3. Modelo de Gestión del Servicio.....	71
4.3.1. Modelo de Relación	72
4.3.2. Comités de Gestión	72
4.3.2.1. Comité de Dirección.....	73
4.3.2.2. Comité de Seguimiento	74



4.3.2.3. Comité Operativo.....	75
4.4. Modelo Económico.....	82
4.4.1. Facturación Base	82
4.4.2. Facturación por <i>Sprint</i>	83
5. PLANIFICACIÓN DEL PFC Y PRESUPUESTO	84
5.1. Planificación del PFC.....	84
5.2. Presupuesto.....	87
6. CONCLUSIONES Y LINEAS FUTURAS.....	89
6.1. Conclusiones.....	89
6.2. Líneas Futuras.....	91
7. BIBLIOGRAFIA.....	93

INDICE DE TABLAS

Tabla 1: Descripción Metodologías Ágiles.....	11
Tabla 2: Gestión de la Demanda	45
Tabla 3: Gestión de Incidencias	47
Tabla 4: Definición del Alcance	49
Tabla 5: Seguimiento	50
Tabla 6: Gestión de Riesgos.....	53
Tabla 7: Ingeniería de Requisitos	55
Tabla 8: Valoración y Planificación del <i>Sprint</i>	57
Tabla 9: Desarrollo Software	60
Tabla 10: Despliegues	62
Tabla 11: Certificación de Equipos de Desarrollo.....	65
Tabla 12: Comité de Dirección	73
Tabla 13: Comité de Seguimiento	74
Tabla 14: Planificación del <i>Sprint</i>	75
Tabla 15: Scrum Diario	76
Tabla 16: Reunión de Revisión del <i>Sprint</i>	78
Tabla 17: Reunión de Retrospectiva del <i>Sprint</i>	79
Tabla 18: Reunión de Certificación.....	81
Tabla 19: Planificación del proyecto	84
Tabla 20: Costes Recursos Humanos.....	87
Tabla 21: Costes Equipos y Herramientas	87
Tabla 22: Otros costes directos asociados al proyecto	88
Tabla 23: Costes indirectos asociados al proyecto.....	88
Tabla 24: Resumen de Costes asociados al proyecto.....	88
Tabla 25: Diferencias entre metodologías ágiles y tradicionales.....	89

INDICE DE FIGURAS

Figura 1: Metodología Ágil Scrum en la Gestión de Proyectos	5
Figura 2: <i>Waterfall</i> vs. <i>Agile</i>	10
Figura 3: Manifiesto Ágil.....	15
Figura 4: Proceso Scrum	20
Figura 5: Comunicación entre los roles de Scrum	20
Figura 6: Artefactos en el flujo de Scrum	23
Figura 7: Eventos en el flujo de Scrum	26
Figura 8: Baraja de cartas <i>Planning Poker</i>	30
Figura 9: Scrum <i>Taskboard</i>	32
Figura 10: Gráfica de Burn Down con avance ideal.....	33
Figura 11: Gráfica de Burn Down habitual	34
Figura 12: Gráfico de Burn Down de un “ <i>Sprint</i> Subestimado”	34
Figura 13: Gráfico de Burn Down de un “ <i>Sprint</i> Sobrestimado”	35
Figura 14: Modelos de un servicio Gestionado	38
Figura 15: Modelo Operativo de un servicio Gestionado	39
Figura 16: Ciclo de Vida de un Proyecto.....	44
Figura 17: Gestión de la Demanda	45
Figura 18: Gestión de Incidencias.....	47
Figura 19: Definición del Alcance	49
Figura 20: Seguimiento.....	50
Figura 21: Ejemplo de Acta de Seguimiento	52
Figura 22: Gestión de Riesgos	53
Figura 23: Ingeniería de Requisitos	55
Figura 24: Valoración y Planificación del <i>Sprint</i>	57
Figura 25: Ejemplo de Lista de Producto.....	59
Figura 26: Desarrollo Software.....	60
Figura 27: Despliegues.....	62
Figura 28: Certificación de Equipos de Desarrollo	64
Figura 29: Modelo Organizativo de un Servicio Gestionado	66



Figura 30: Organización de los Equipos de Trabajo	67
Figura 31: Modelo de Gestión de un Servicio Gestionado	71
Figura 32: Modelo de Relación entre Comités	72
Figura 33: Modelo Económico de un Servicio Gestionado	82
Figura 34: Gantt del proyecto	86

1. INTRODUCCIÓN

1.1. Introducción

A día de hoy las empresas se encuentran en un entorno de trabajo dinámico y exigente, la alta competitividad entre ellas hace que evolucionen de forma rápida y adopten nuevas estrategias.

Las organizaciones necesitan enfrentarse a los cambios del mercado; la aparición de nuevos productos y servicios con ciclos de vida más cortos, obliga a incrementar la productividad y disminuir el tiempo de reacción, adaptarse rápidamente a las variantes necesidades de los clientes y, en definitiva, aumentar la capacidad de competir en un mercado mucho más amplio.

La presión de las organizaciones por obtener resultados rápidos y tangibles, así como su exigencia para demandar cambios sobre dichos resultados, hace también que las metodologías de trabajo se adapten. En este contexto aparecen las metodologías ágiles como una opción pues, en contraposición con las metodologías convencionales que actúan basadas en principios de estabilidad y control del contexto, las metodologías ágiles no necesitan tener un sistema perfectamente definido como paso previo a su desarrollo, sino que perciben cada respuesta al cambio como una oportunidad para mejorar e incrementar la satisfacción del cliente, considerando la gestión de cambios como un aspecto inherente al propio proceso de desarrollo y, permitiendo de este modo, una mejor adaptación.

Las metodologías ágiles no pueden ser aplicadas a toda clase de proyectos por lo que conviven las dos corrientes, metodologías clásicas o métodos predictivos y metodologías ágiles.

Actualmente, la metodología ágil más popular para la gestión de proyectos es Scrum. Se presenta como contrapunto a PMBOK y PRINCE2, siendo utilizada tanto para desarrollo de software como para otro tipo de productos. Por otra parte, también se disponen de otras metodologías ágiles como *Dynamic Systems Development Method* (DSDM), metodología ágil más veterana y más próxima a los métodos tradicionales; *Extreme Programming* (XP), metodología ágil más radical y popular, XP se centra en el ciclo de vida del desarrollo de software; *Agile Modeling*, metodología para el modelado y la generación de documentación que se encuentra alineado con los principios del desarrollo ágil; *Feature Driven Development* (FDD), metodología de desarrollo de software orientada a la generación de valor para el cliente; *Crystal Methodologies*, conjunto de metodologías para el desarrollo de software; *Adaptive Software Development* (ASD), método a través del cual inculcar una cultura adaptativa a la empresa; *Dynamic Systems Development Method* (DSDM), metodología RAD (Desarrollo rápido de aplicaciones) unificada. [1]

Si bien las metodologías tradicionales de desarrollo de software presentan modelos de procesos que cubren todas las necesidades de los sistemas de información, las metodologías ágiles listadas abarcan áreas complementarias entre sí, por lo que es posible aplicar varias metodologías en conjunto a la vez para un mismo proyecto.

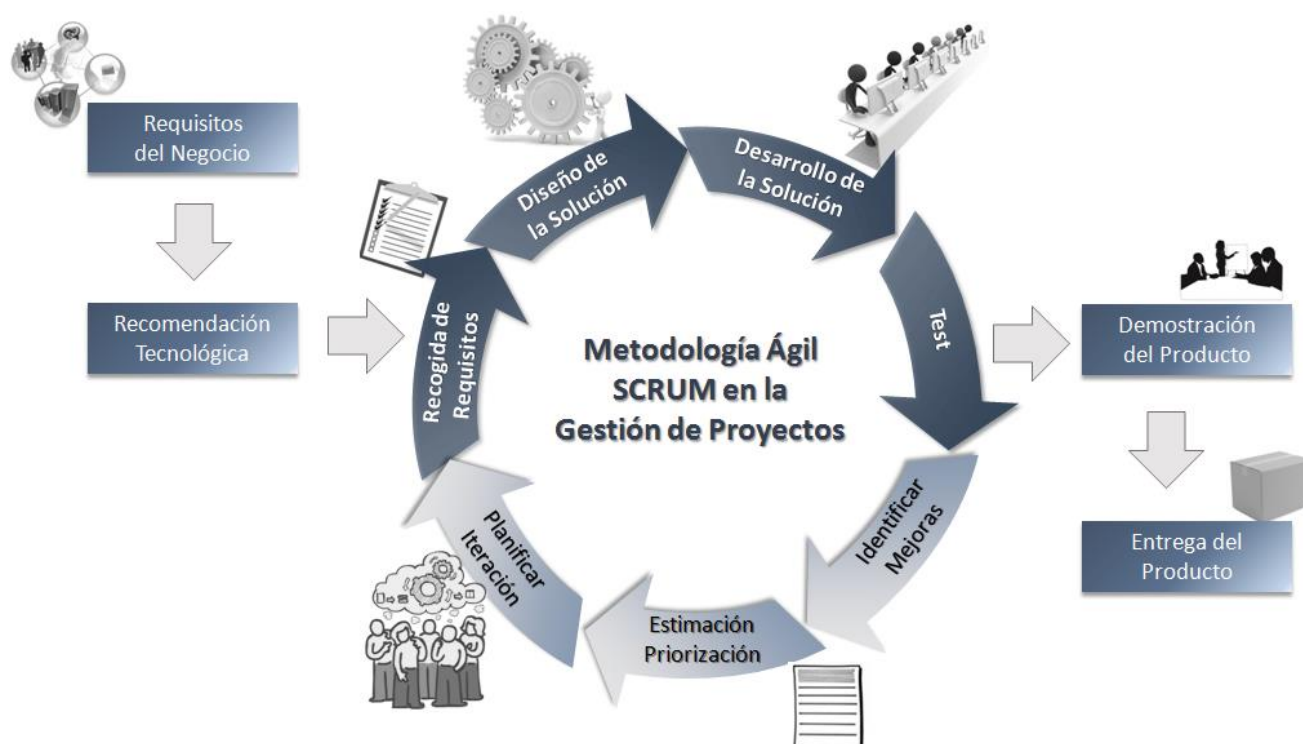


Figura 1: Metodología Ágil Scrum en la Gestión de Proyectos

1.2. Objetivo

El objetivo principal de este proyecto es presentar las Metodologías Ágiles aplicadas a la gestión de proyectos y en concreto Scrum, metodología que está teniendo especial éxito. Se aplicará el método Scrum a un caso práctico para analizar cómo es su desarrollo y los beneficios que aporta.

Para ello, inicialmente se hará un repaso de la historia de los procesos para la gestión de proyectos analizando cómo surgen las metodologías ágiles y cuáles son las más destacadas. Se estudiará las diferencias con las metodologías convencionales y se elaborará una guía sencilla que ayude a decidir en un proyecto por usar una metodología ágil o convencional, no se tratará de elegir un modelo como el mejor, simplemente habrá casos en los que convendrá una gestión predictiva y otros en los que la opción ágil puede ser más beneficiosa.

Se expondrá la metodología ágil Scrum, los elementos y roles que en ella intervienen, y cómo es su proceso y ciclo de vida. Se describirán también algunas de las herramientas que se utilizan en su desarrollo.

Una vez adquiridos los conocimientos básicos sobre las metodologías ágiles y en concreto Scrum, se empleará dicha metodología en un proyecto real de un Cliente del Sector Público que desea externalizar uno de sus servicios al ciudadano. El proyecto detallará los servicios ofrecidos describiendo los recursos necesarios y las acciones a llevar a cabo, además de especificar cómo actúa cada uno de los roles de Scrum y el papel que desempeñan dentro del proyecto.

1.3. Motivación personal

Dado el auge que están teniendo las metodologías ágiles en el sector de las TI, no sólo para los proyectos de desarrollo de software, sino en la gestión de proyectos como el expuesto en el caso práctico. La motivación personal del presente proyecto fin de carrera (PFC) es conocer la aplicación de dichas metodologías y en especial Scrum, metodología que están implantando clientes de mi organización en sus proyectos.

Aunque no corresponde a mi actual desarrollo profesional, en el mundo de la tecnología es esencial estar actualizado e informado de los últimos avances debido a que la evolución es rápida y la adquisición de responsabilidades así lo exige.

En este PFC trabajaré junto con mis clientes y estudiaré el desarrollo de un proyecto completo desde una visión global, ampliando la implicación parcial que actualmente ocupo en mi organización desde el rol que desempeño.

1.4. Estructura de la memoria

La memoria se estructura en diferentes capítulos tal y como se detalla a continuación:

1. *Introducción*: Capítulo actual, dedicado a la explicación e introducción al contenido del proyecto. Se realiza una breve introducción del marco en el que se presenta y se concreta el objetivo así como la estructura de la memoria.
2. *Metodologías Ágiles de Gestión de Proyectos*: Se procede al estudio de la evolución de los procesos de desarrollo pasando a describir las metodologías ágiles más destacadas actualmente. Se propone una guía sencilla para determinar cuándo usar una metodología ágil, y el capítulo finaliza con la descripción del Manifiesto Ágil firmado en febrero de 2001.
3. *Scrum*: Se detalla las principales características de la metodología ágil Scrum analizando los elementos que componen su marco de trabajo: Roles, Componentes, Eventos y Herramientas. Antes de finalizar el capítulo se resume el proceso de Scrum viendo cómo interactúan cada uno de los elementos anteriormente descritos.
4. *Caso de Estudio*: En este capítulo se pone en práctica la metodología ágil Scrum considerando un proyecto real. Se especificarán las tareas necesarias para realizar la puesta en marcha de un servicio gestionado en un cliente distinguiendo los diferentes elementos de Scrum.

Para realizar el caso de estudio se ha utilizado además de las herramientas básicas de ofimática, el programa informático *Visio 2013 de Microsoft*, apto para realizar los diagramas de flujo.



5. *Planificación del PFC y Presupuesto:* Se define el plan llevado a cabo para la elaboración de este documento técnico y el coste asociado al proyecto.
6. *Conclusiones y Líneas Futuras:* Contiene las conclusiones obtenidas del trabajo llevado a cabo y los posibles campos de ampliación del presente proyecto.
7. *Bibliografía:* Se enumeran las referencias utilizadas en la elaboración de este proyecto, tanto bibliográficas como digitales.

2. METODOLOGÍAS ÁGILES DE GESTIÓN DE PROYECTOS

2.1. Historia de los procesos de desarrollo

Cuando aparecieron las primeras computadoras digitales en la década de los cuarenta [2], el desarrollo de software era algo tan nuevo que era casi imposible hacer predicciones de las fechas estimadas de finalización del proyecto, y muchos de ellos sobrepasaban los presupuestos y tiempo estimados. Los desarrolladores tenían que volver a escribir todos sus programas para correr en máquinas nuevas que salían cada uno o dos años, haciendo obsoletas las ya existentes. El término Ingeniería del software apareció por primera vez a finales de la década de los cincuenta para ayudar a identificar y corregir mediante principios y metodologías los procesos de desarrollo y mantenimiento de sistemas de software.

Un objetivo de décadas ha sido el encontrar procesos y metodologías, que sean sistemáticas, predecibles y repetibles, a fin de mejorar la productividad en el desarrollo y la calidad del producto software. Estas metodologías o modelos definen el proceso que se debe seguir para construir, entregar y hacer evolucionar el software, desde la concepción de una idea hasta la entrega y la retirada del sistema, y representa todas las actividades y artefactos (productos intermedios) necesarios para desarrollar una aplicación. [3]

En la década de los sesenta aparece el denominado modelo en cascada, que sirvió como base para la formulación del análisis estructurado, el cual fue uno de los precursores en este camino hacia la aplicación de prácticas estandarizadas.

El modelo en cascada surge como respuesta al modelo 'codificar y probar'. En esa época ya existían modelos iterativos e incrementales pero no eran disciplinados ni estaban formalizados. A consecuencia de esta realidad, la idea de tener un modelo que ordenara el proceso de desarrollo y que parecía bastante sencillo de llevar a la práctica y de comunicar hizo que el modelo en cascada tuviera una gran promoción.

Este modelo ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior [4]. Cualquier error de diseño detectado en la etapa de pruebas conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo. La palabra cascada sugiere, mediante la metáfora de la fuerza de la gravedad, el esfuerzo necesario para introducir un cambio en las fases más avanzadas de un proyecto.

Dadas las características inherentes del modelo, la fase de implementación del mismo requiere el desarrollo de los módulos de forma independiente con las correspondientes pruebas unitarias, y en la siguiente fase, se realiza la integración de los mismos. Este modelo generaba grandes inconvenientes debidos a que todo estaba probado de forma unitaria sin interacción con los demás módulos. Las sorpresas llegaban cuando se integraban estas piezas para formar la aplicación; lo cual inevitablemente desembocaba en un retraso del proyecto, sacrificando la calidad del mismo.

De esta forma fueron surgiendo diversos procesos denominados iterativos que proponían lidiar con los errores del modelo en cascada, mitigando los riesgos de forma temprana. Algunos de estos procesos son el modelo iterativo e incremental, el modelo en espiral, el modelo basado en prototipo, el modelo RUP¹, etc.

Básicamente, el pilar de estos modelos es basar el desarrollo en iteraciones e ir construyendo la aplicación de forma progresiva, agregando funcionalidad sucesivamente. Las iteraciones representan un mini-proyecto autocontenido el cual está compuesto por todas las fases del desarrollo (requerimientos, diseño, implementación, *testing*). Los incrementos están dados por la funcionalidad que se va agregando de forma iterativa. Gracias a estas iteraciones se logra entre otras cosas obtener el *feedback* necesario del cliente que era frenado en el modelo en cascada una vez se finalizaba la fase de requerimientos. Consecuentemente podemos argumentar que los modelos iterativos fomentan el cambio de forma temprana y proponen un control de cambio disciplinado que permita que el usuario ajuste sobre el desarrollo sus requerimientos. Esto se contrapone a la intolerancia del modelo en cascada para lidiar con dichos cambios.

Del modelo en espiral desarrollado por Barry Boehm [5] surgió una de las ideas fundamentales que las metodologías posteriores adoptarían: el temprano análisis de riesgos. El modelo en espiral, de carácter iterativo en sus primeras fases, plantea la necesidad de realizar al principio diversas iteraciones dirigidas a mitigar los riesgos más críticos relevados en el proyecto mediante la realización de prototipos o simulaciones de tipo desechables tendientes a probar algún concepto. Una vez que esos prototipos son validados se suceden iteraciones del tipo: determinar objetivos, evaluar, desarrollar, planear. Una vez que se tenía el diseño detallado y validado por el cliente, se implementaba el software siguiendo las etapas de un modelo en cascada. Esta es una falla importante del modelo ya que no se acomoda a la posibilidad de cambios una vez que se inicia la construcción. Todas las críticas que se le hacían al modelo en cascada se aplican a estas fases del modelo en espiral.

Fue el mismo Barry Boehm, autor de este modelo, quien en su artículo [6] describe tres hitos críticos a ser utilizados en cualquier proyecto para poder planificar y controlar el progreso del mismo, dando visibilidad a los afectados *stakeholders*. Estos hitos están relacionados con las etapas de avance que se van dando a lo largo de un proyecto de acuerdo a como ocurren las actividades de ingeniería (que componen los espirales del modelo en espiral) a las actividades de producción (que componen la construcción en cascada del software). Su impacto en la industria del software ha sido tan importante que uno de los procesos más utilizados en la actualidad, el RUP, los incorpora. Estos hitos son:

- Objetivos del Ciclo de Vida

Definición del alcance del software a construir, la identificación de los *stakeholders*, y el delineamiento del plan de desarrollo del sistema

¹ RUP (*Rational Unified Process*)

- Arquitectura del Ciclo de Vida

Delineamiento de la arquitectura del sistema, resolución de todos los riesgos críticos del proyecto, y refinamiento de los objetivos y el alcance del sistema.

A partir de este hito, se comienza la construcción en forma masiva del sistema, llegándose a utilizar el máximo de recursos en el proyecto. Asimismo, comienzan las fases más predecibles en cierta medida del desarrollo.

- Capacidad Operativa Inicial

Entrega del primer *release* del software, incorpora la funcionalidad definida en la correspondiente iteración. También se espera tener material de entrenamiento, como un Manual del Usuario y Manual de Operaciones.

Estos hitos propuestos por Boehm son independientes del proceso de desarrollo elegido y permiten una estandarización de entregas. Para los clientes, los hitos otorgan visibilidad sobre el proyecto pudiendo medir el progreso. Para el equipo de desarrollo, los hitos proveen una guía de las fases del proyecto orientada a los entregables necesarios para cada hito así como la posibilidad de recibir *feedback* de los clientes/usuarios sobre los productos que son entregados en el tiempo.

Con este modelo en espiral en el que cada una de las fases de cascada termina con una evaluación de riesgos y un prototipo, permite a los usuarios determinar si el proyecto continúa, debe volver a fases anteriores, o debe terminar. Sin embargo, las fases son todavía lineales, los requisitos se realizan en la fase de requisitos, el diseño en la fase de diseño, y así sucesivamente. Por este motivo surgen las denominadas Metodologías Ágiles que asumen que el análisis, diseño, etc., de cada iteración son impredecibles, es decir, no son (o a priori no tienen porque) lineales y son flexibles, tal y como se muestra en la Figura 2.

Modelo en Cascada (*Waterfall*)



Metodologías Ágiles (*Agile*)

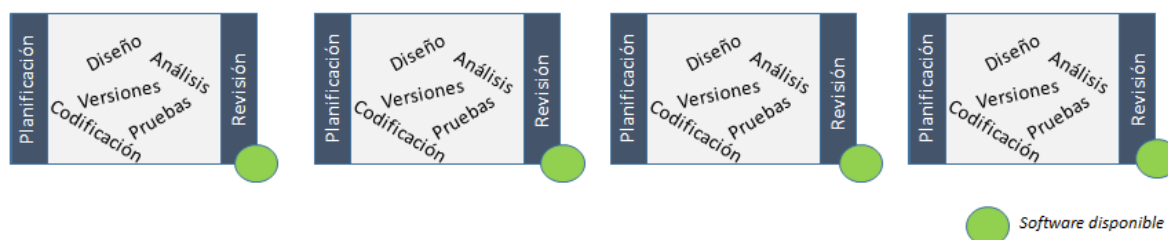


Figura 2: *Waterfall* vs. *Agile*

2.2. Metodologías Ágiles

Hoy en día las organizaciones de desarrollo de aplicaciones hacen uso de las dos grandes aproximaciones para la metodología en el desarrollo de un proyecto. La que tradicionalmente se conoce como “desarrollo en cascada” y las nuevas metodologías que proponen la generación de pequeños entregables en un esquema de actividades que se pueden solapar o superponer, ya sea en forma secuencial o con un enfoque totalmente solapado, estas son las llamadas Metodologías Ágiles. Se debe puntualizar que las metodologías ágiles no inventaron la noción de los procesos iterativos e incrementales, los cuales eran usados desde décadas pasadas incluso en momentos en que el modelo en cascada era el estándar.

Las Metodologías Ágiles o *Agile* en inglés se basan en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos auto-organizados y multidisciplinarios. Existen muchos métodos de desarrollo ágil; la mayoría minimizan riesgos desarrollando software en lapsos cortos. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, sino que la meta es tener una «demo» (sin errores) al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

Este tipo de metodologías requieren que todo el equipo de desarrollo esté involucrado en todas las etapas del proyecto en constante contacto con el usuario final de la aplicación. Los requerimientos *Agile* son representados como “Historias de Usuario” (*Userstories*), las cuales son una pequeña descripción de los requerimientos del usuario descrito en un lenguaje lo más aproximado a sus jerga diaria. Se hace un énfasis muy fuerte a que este tipo de comunicación esté orientada al usuario final y que sea fácil de entender. Estas historias definen alcances fáciles de identificar para poder efectuar planes iterativos para definición, revisión y confirmación de alcances. No se le exige al cliente o usuario final que tenga que conocer desde el principio todos sus requerimientos. Sin embargo, sí se pide que no se cambien en los ciclos iterativos (*Sprints*) que se definen para el desarrollo del producto.

Entre las metodologías ágiles más destacadas hasta el momento se pueden identificar:

Metodologías Ágiles	Descripción
XP – Extreme Programming	<p>La programación extrema o <i>eXtreme Programming</i> (XP) es un enfoque de la ingeniería de software formulado por Kent Beck [7]. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.</p> <p>XP o <i>eXtreme Programming</i> (XP) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los</p>

	<p>desarrolladores, y propiciando un buen clima de trabajo.</p> <p>XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. Permite a las personas relacionarse y comunicarse para encontrar soluciones, sin jerarquías ni enfrentamientos.</p> <p>XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.</p>
Scrum	<p>Scrum es un marco de trabajo (<i>framework</i>) para mejorar la forma en la que trabajan las personas, o como lo define la <i>Scrum Alliance</i>: "un <i>framework</i> basado en equipos para desarrollar sistemas y productos complejos" [8].</p> <p>Scrum utiliza un proceso iterativo en donde cada iteración (llamadas <i>Sprints</i>) es lo más corta posible, manteniendo un ritmo parejo a medida que se avanza con planificación, ejecución y reflexión.</p> <p>En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.</p> <p>Scrum requiere un foco claro, compromiso y transparencia completa en todos los niveles; adopta y enfatiza ciertos valores humanos incluyendo (pero no limitado) a la confianza, la integridad, el coraje y el respeto.</p>
Crystal Methodologies	<p><i>Crystal Methodologies</i> se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. El equipo es la base de estas metodologías creadas por Alistair Cockburn.</p> <p>El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo.</p> <p><i>Crystal</i> establece una serie de políticas de trabajo en equipo (<i>Methods</i>) orientadas a fomentar la mejora de estas habilidades. Dependiendo del tamaño del equipo, se establece una metodología u otra designadas por color: <i>Crystal Clear</i> (para 3-8 personas), <i>Crystal Yellow</i> (para 10-20 personas), <i>Crystal Orange</i> (para 25-50), etc.</p>

<p>DSDM – Dynamic Systems Development Method</p>	<p>Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD (Desarrollo rápido de aplicaciones) unificada.</p> <p>Es un proceso iterativo e incremental donde el equipo de desarrollo y el usuario trabajan juntos.</p> <p>Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.</p>
<p>FDD – Feature Driven Development</p>	<p>Enfoque ágil para el desarrollo de sistemas, sus impulsores son Jeff De Luca y Peter Coad. El enfoque no hace énfasis en la obtención de los requerimientos sino en cómo se realizan las fases de diseño y construcción. Sin embargo, fue diseñado para trabajar con otras actividades de desarrollo de software y no requiere la utilización de ningún modelo de proceso específico.</p> <p>Se basa en un ciclo muy corto de iteración, nunca superior a dos semanas, y en el que el análisis y los desarrollos están orientados a cumplir una lista de características (<i>features</i>) que tiene que tener el software a desarrollar.</p> <p>La metodología sigue cinco fases iterativas: desarrollo de un modelo global, construcción de una lista de características, planificación, diseño de la característica e implementación de la característica.</p>
<p>ASD – Adaptive Software Development</p>	<p>Es una metodología de desarrollo impulsada por Jim Highsmith y Sam Bayer que hace énfasis en aplicar las ideas que se originaron en el mundo de los sistemas complejos. En realidad no es una metodología de desarrollo de software, sino un método a través del cual inculcar una cultura adaptativa a la empresa.</p> <p>Es un modelo iterativo, orientado a la gestión del cambio y facilitando la colaboración y la interacción de las personas a nivel interpersonal, cultural y estructural. Marca una estrategia de desarrollo rápido pero con rigor y disciplina.</p>

Tabla 1: Descripción Metodologías Ágiles

2.3. ¿Cuándo se aplica una Metodología Ágil?

No siempre se debe utilizar una metodología ágil, dependerá de cada proyecto en concreto. Lo que un proyecto busca es el éxito y para ello necesitará una metodología adecuada a él que lo garantice adaptándose a los objetivos buscados, al equipo de trabajo, a los recursos disponibles, al plazo de entrega, al entorno socio-cultural, etc.

Las siguientes preguntas se pueden utilizar como guía para discernir si se debe aplicar una metodología tradicional o una metodología ágil:

1. ¿Cuál es el tamaño del proyecto?

Si es muy grande y/o involucra a muchas personas en el equipo, lo mejor es que se utilicen metodologías más estrictas y que se basen en la planificación y control del proyecto. En cambio, si se trata de un proyecto pequeño y/o con un equipo reducido de desarrollo, de tres a ocho personas, es una buena oportunidad para los métodos ágiles.

2. ¿Los requisitos del proyecto son dinámicos?

En entornos donde los requisitos son cambiantes o poco definidos las metodologías ágiles son las más adecuadas. Si el ámbito no es dinámico las metodologías orientadas a un plan que contempla cambios previsibles se ajustará mejor al proyecto.

3. ¿Qué ocurre cuando el sistema falla?

Si el sistema es crítico en el que cualquier fallo involucra pérdidas humanas o grandes cantidades de dinero será conveniente usar métodos clásicos que garanticen la calidad y funcionen a la primera.

4. ¿Cuál es la relación con el cliente?

Para utilizar una metodología ágil es necesario que el cliente esté involucrado en el desarrollo del proyecto y se mantenga el contacto continuo. Una metodología clásica necesitará un gran análisis inicial.

5. ¿Cómo es el equipo de desarrollo?

Las metodologías ágiles requieren de una gran madurez, experiencia y gran talento. Deben ser equipos con gente *seniors* o *semi-seniors*.

6. ¿Cuál es la cultura empresarial de la empresa en la que se va a desarrollar el proyecto?

Se requiere un cierto ambiente "informal", que fomente la comunicación de igual a igual, para hacer uso de una metodología ágil. Si la organización en la que se quiere desarrollar el proyecto

tiene un alto grado de ceremonia y jerarquías estrictas, es conveniente usar métodos tradicionales.

Si el equipo es pequeño y está formado mayoritariamente por gente con talento y experiencia, si el cliente final está involucrado y no impone barreras de comunicación, si los requisitos son altamente cambiantes, si no es un proyecto crítico y no es demasiado grande, es el mejor escenario para implementar el proyecto con una metodología ágil.

2.4. Manifiesto Ágil

El 17 de febrero de 2001 diecisiete críticos² de los modelos de mejora del desarrollo de software basado en procesos, convocados por Kent Beck, se reunieron en Snowbird, Utah para tratar sobre técnicas y procesos para desarrollar software. En la reunión se identificó el término “Métodos Ágiles” para definir a los métodos que estaban surgiendo como alternativa a las metodologías formales a las que consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo.

Los integrantes de la reunión resumieron la filosofía sobre la que se basan los métodos ágiles en cuatro postulados y doce principios, lo que ha quedado denominado como Manifiesto Ágil [9]:

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

- *Individuos e iteraciones sobre procesos y herramientas*
- *Software funcionando sobre documentación extensiva*
- *Colaboración con el cliente sobre negociación contractual*
- *Respuesta ante el cambio sobre seguir un plan*

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Figura 3: Manifiesto Ágil

² Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland y Dave Thomas

- Valoramos más a los individuos y su interacción que a los procesos y las herramientas.

Este es el valor más importante del manifiesto.

Por supuesto que los procesos ayudan al trabajo. Son una guía de operación. Las herramientas mejoran la eficiencia, pero hay tareas que requieren talento y necesitan personas que lo aporten y trabajen con una actitud adecuada.

La producción basada en procesos persigue que la calidad del resultado sea consecuencia del *know-how* “explicitado” en los procesos, más que en el conocimiento aportado por las personas que los ejecutan.

Sin embargo en desarrollo ágil los procesos son una ayuda. Un soporte para guiar el trabajo. La defensa a ultranza de los procesos lleva a afirmar que con ellos se pueden conseguir resultados extraordinarios con personas mediocres, y lo cierto es que este principio no es cierto cuando se necesita creatividad e innovación.

- Valoramos más el software que funciona que la documentación exhaustiva.

Poder anticipar cómo será el funcionamiento del producto final, observando prototipos previos, o partes ya elaboradas ofrece un “*feedback*” estimulante y enriquecedor, que genera ideas imposibles de concebir en un primer momento, y difícilmente se podrían incluir al redactar un documento de requisitos detallado en el comienzo del proyecto.

El manifiesto ágil no da por inútil la documentación, sólo la de la documentación innecesaria. Los documentos son soporte de hechos, permiten la transferencia del conocimiento, registran información histórica, y en muchas cuestiones legales o normativas son obligatorios, pero su relevancia debe ser mucho menor que el producto final.

La comunicación a través de documentos no ofrece la riqueza y generación de valor que logra la comunicación directa entre las personas, y a través de la interacción con prototipos del producto. Por eso, siempre que sea posible debe preferirse reducir al mínimo indispensable el uso de documentación, que requiere trabajo sin aportar un valor directo al producto. Si la organización y los equipos se comunican a través de documentos, además de ocultar la riqueza de la interacción con el producto, forman barreras de burocracia entre departamentos o entre personas.

- Valoramos más la colaboración con el cliente que la negociación contractual.

Las prácticas ágiles están indicadas para productos cuyo detalle resulta difícil prever al principio del proyecto; y si se detallara al comenzar, el resultado final tendría menos valor que si se mejoran y precisan con retroinformación continua durante él.

También son apropiadas cuando se prevén requisitos inestables por la velocidad de cambio en el entorno de negocio del cliente. El objetivo de un proyecto ágil no es controlar la ejecución conforme a procesos y cumplimiento de planes, sino proporcionar el mayor valor posible al producto.

Resulta por tanto más adecuada una relación de implicación y colaboración continua con el cliente, más que una contractual de delimitación de responsabilidades.

- Valoramos más la respuesta al cambio que el seguimiento de un plan

Para desarrollar productos de requisitos inestables, que tienen como factor inherente el cambio y la evolución rápida y continua, resulta mucho más valiosa la capacidad de respuesta que la de seguimiento y aseguramiento de planes. Los principales valores de la gestión ágil son la anticipación y la adaptación, diferentes a los de la gestión de proyectos ortodoxa: planificación y control que evite desviaciones del plan.

Los doce principios del Manifiesto Ágil son:

- I. *Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.*
- II. *Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.*
- III. *Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.*
- IV. *Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.*
- V. *Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.*
- VI. *El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.*
- VII. *El software funcionando es la medida principal de progreso.*
- VIII. *Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.*
- IX. *La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.*
- X. *La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.*
- XI. *Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.*
- XII. *A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para la continuación de ajustar y perfeccionar su comportamiento en consecuencia.*

El Manifiesto para el Desarrollo Ágil de Software es de suma importancia dentro del movimiento de las metodologías ágiles. El mismo representa una iniciativa conjunta entre los principales responsables de los procesos ágiles para lograr unificar principios compartidos por las diversas metodologías de manera de crear un *framework* que contribuya al mejoramiento del desarrollo ágil.

3. SCRUM

3.1. Introducción

El concepto de Scrum tiene su origen en un estudio de 1986 [10] en el que Ikujiro Nonaka e Hirotaka Takeuchi analizaron cómo grandes empresas tecnológicas de Japón y Estados Unidos desarrollaban los nuevos productos; analizaron productos como cámaras de fotos de Canon, fotocopadoras de Xerox, automóviles de Honda u ordenadores de HP entre otros. Los equipos que desarrollaron estos productos partían de requisitos muy generales, así como novedosos, y debían salir al mercado en mucho menos tiempo del que se tardó en lanzar productos anteriores. Estos equipos seguían patrones de ejecución de proyecto muy similares. En este estudio se comparaba la forma de trabajo de estos equipos altamente productivos y multidisciplinarios con la colaboración entre los jugadores de Rugby, por ello su nombre no corresponde a una sigla, sino a un concepto deportivo propio del rugby, relacionado con la formación requerida para la recuperación rápida del juego ante una infracción menor.

En 1995 Ken Schwaber y Jeff Sutherland formalizaron el proceso conocido como Scrum, desde entonces tanto Sutherland como Schwaber han sido referentes para Scrum y han seguido colaborando en la publicación de documentos sobre la metodología. Finalmente, tras la formulación del Manifiesto Ágil en el 2001, varias metodologías, entre ellas Scrum, se agruparon bajo la categoría de Desarrollo Ágil de Software. En la actualidad, Scrum se está utilizando en diferentes tipos de negocio. La *Scrum Alliance* es la organización sin ánimo de lucro que se encarga de difundir Scrum en este ámbito [7].

Scrum es un marco de procesos utilizado para gestionar el desarrollo de productos complejos. Scrum no es un proceso o una técnica para construir productos; más bien, es un marco en el que se pueden emplear diferentes procesos y técnicas para obtener el mejor resultado posible de un proyecto.

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo afirma que el conocimiento proviene de la experiencia de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la previsibilidad del riesgo y control. Esta teoría se fundamenta en tres pilares:

- Transparencia:

La transparencia garantiza la visibilidad de las cosas que pueden afectar al resultado; La transparencia requiere que los aspectos del proceso sean definidos por un estándar común, de modo que los observadores compartan un entendimiento común de lo que se está viendo.

- Inspección:

La inspección ayuda a detectar variaciones no deseables en el proceso. Se debe inspeccionar frecuentemente los componentes y el progreso hacia el objetivo pero la frecuencia no debe ser tan alta que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan con diligencia por inspectores expertos en el lugar de trabajo.

- Adaptación:

La adaptación para realizar los ajustes pertinentes que un inspector determine, estos ajustes deben realizarse cuanto antes para minimizar el impacto de los mismos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

El marco de trabajo Scrum consiste en los siguientes elementos: equipo Scrum, roles, eventos, componentes y reglas asociadas. Cada elemento dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso [11].

En la figura 4 podemos observar los elementos comentados que pasaremos a describir en los siguientes apartados.

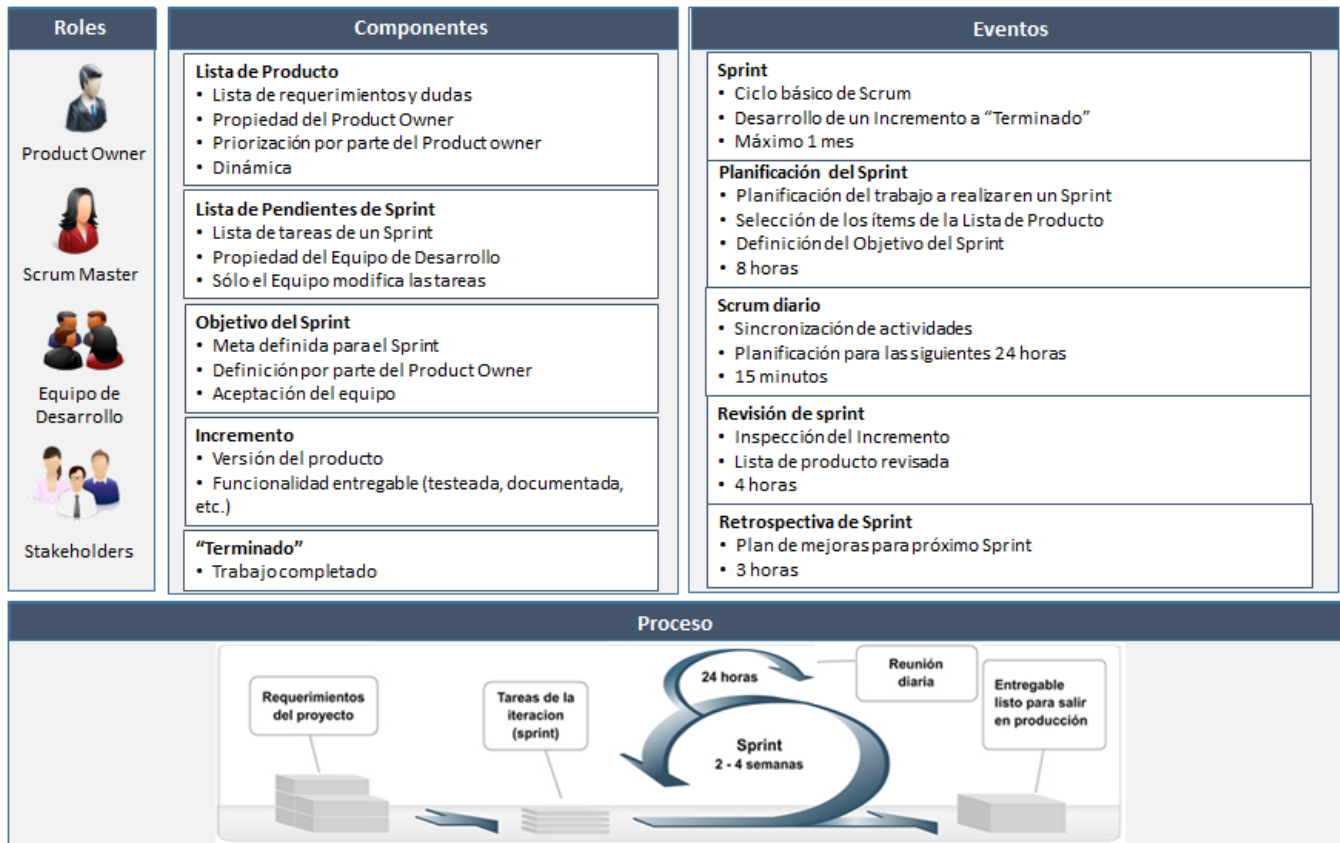


Figura 4: Proceso Scrum

3.2. Roles de Scrum

Los Equipos Scrum son autoorganizados y multifuncionales. No son dirigidos por personas externas al trabajo y el equipo es autosuficiente para realizar su objetivo, entre ellos deciden cuál es la mejor manera de llevar a cabo el trabajo.

El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

Dentro del Equipo Scrum se definen los siguientes roles:



Figura 5: Comunicación entre los roles de Scrum

3.2.1. *Product Owner* (Propietario del Producto)

El *Product Owner* es la persona que representa al cliente/usuario final del producto (solamente una persona para eliminar las posibles confusiones o interferencias). Es el responsable de escribir las historias de usuario (descripción de los requerimientos), priorizarlas y colocarlas en la Lista de Producto o *Product Backlog*.

La gestión de la Lista de Producto incluye:

- Expresar claramente los elementos de la Lista de Producto;
- Ordenar los elementos en la Lista de Producto para alcanzar los objetivos y misiones de la mejor manera posible;
- Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo o *Development Team*;
- Asegurar que la Lista de Producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación; y,
- Asegurar que el Equipo de Desarrollo entiende los elementos de la Lista de Producto al nivel necesario. Estará en constante contacto con el equipo.

El *Product Owner* repartirá los objetivos/requisitos en iteraciones y establecerá un calendario de entregas. Antes de iniciar cada iteración replanificará el proyecto en función de los requisitos que aportan más valor en ese momento, de los requisitos completados en la iteración anterior y del contexto del proyecto en ese momento (demandas del mercado, movimientos de la competencia, etc.).

Para que el *Product Owner* pueda hacer bien su trabajo, toda la organización debe respetar sus decisiones.

3.2.2. Equipo de Desarrollo (*Development Team*)

Son las personas que forman el grupo de desarrollo del producto, entre tres y nueve personas. Tienen la función de entregar un Incremento de producto "Terminado", que potencialmente se pueda poner en producción, al final de cada iteración; en Scrum los ciclos iterativos se llaman *Sprint*. Sólo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

El Equipo de Desarrollo tiene las siguientes características:

- Son autoorganizados. Nadie (ni siquiera el *Scrum Master*) indica al Equipo de Desarrollo cómo convertir elementos de la Lista de Producto en Incrementos de funcionalidad potencialmente desplegables;

- Son multifuncionales, se compone de personas con diferentes perfiles para abarcar todas las posibles áreas y tecnologías del proyecto, aunque la responsabilidad recae en el Equipo de Desarrollo como un todo;
- Scrum no reconoce títulos para los miembros de un Equipo de Desarrollo, todos son Desarrolladores, independientemente del trabajo que realice cada persona; no hay excepciones a esta regla; y,
- Scrum no reconoce sub-equipos, no importan los dominios particulares que requieran ser tenidos en cuenta, como pruebas o análisis de negocio; no hay excepciones a esta regla;

3.2.3. *Scrum Master* (Facilitador)

Es la persona encargada de gestionar el proyecto cuyo principal objetivo es garantizar que las etapas de Scrum se cumplen y son entendidas. Para ello deberá salvar los problemas que puedan surgir, evitar cuellos de botella, fomentar la motivación del equipo de trabajo y tratar que el clima de trabajo sea el adecuado sin intervenciones externas. Será el portavoz del Equipo, atendiendo todo aquello que venga de fuera y transmitiendo todo lo que el Equipo desee.

En los equipos donde el *Scrum Master* no es miembro del Equipo de Desarrollo se le delegan a éste las funciones de actualización de la Lista de Pendientes del *Sprint* o *Sprint Backlog* así como la creación de documentos para las Reuniones de Revisión del *Sprint* y de Retrospectiva del *Sprint*. Quien realice este rol puede realizar también un rol del Equipo de Desarrollo, pero nunca podrá ser *Product Owner* a la vez que *Scrum Master*. El *Scrum Master* facilitará los eventos de Scrum según se requiera o se necesite.

El *Scrum Master* ayuda al *Product Owner* a encontrar técnicas para gestionar la Lista de Producto de manera efectiva así como a entender la necesidad de contar con elementos de la Lista de Producto claros y concisos;

El *Scrum Master* también dará servicio a la organización de varias formas, incluyendo:

- Liderar y guiar a la organización en la adopción de Scrum;
- Planificar las implementaciones de Scrum en la organización;
- Ayudar a los empleados e interesados a entender y llevar a cabo Scrum y el desarrollo empírico de producto;
- Motivar cambios que incrementen la productividad del Equipo Scrum; y,
- Trabajar con otros *Scrum Masters* para incrementar la efectividad de la aplicación de Scrum en la organización.

3.3. Componentes (*Artefacts*)

Los componentes maximizarán la transparencia de la información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del componente.

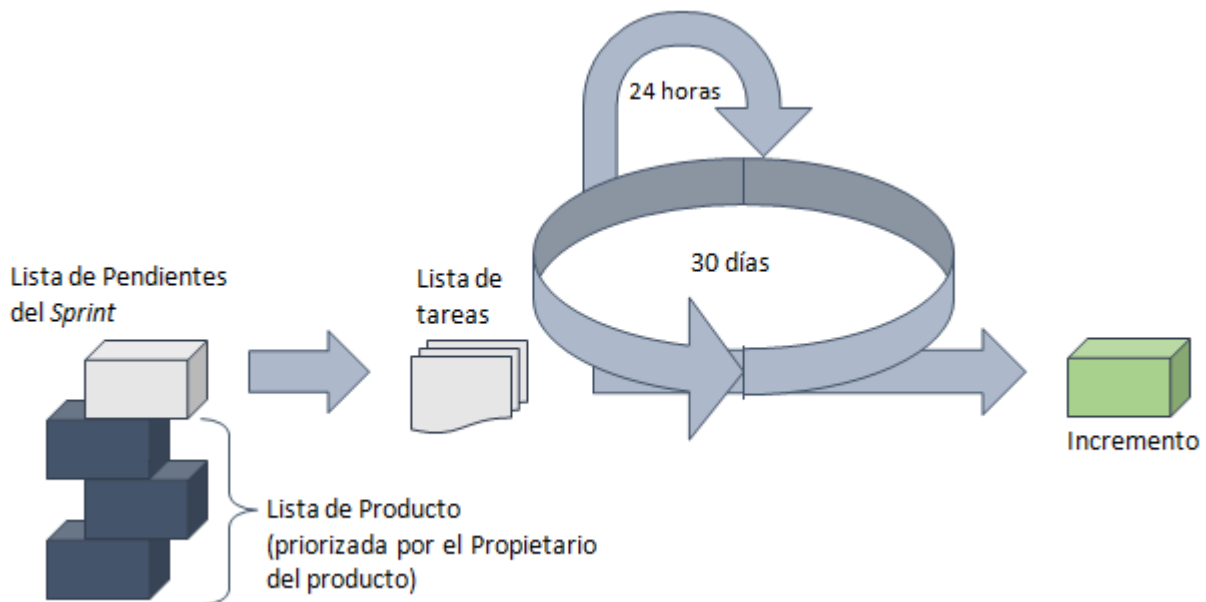


Figura 6: Artefactos en el flujo de Scrum

3.3.1. Lista de Producto (*Product Backlog*)

La Lista de Producto recoge todos los requisitos necesarios para formar el producto. Su contenido, disponibilidad y ordenación es responsabilidad del *Product Owner*.

La Lista de Producto es dinámica, es decir, evolucionará a medida que el producto y el entorno en el que se está usando también cambien; de esta manera se puede identificar lo que el producto necesita para ser adecuado, competitivo y útil. Los cambios en los requisitos de negocio, las condiciones del mercado o la tecnología podrían causar cambios en la Lista de Producto.

La Lista de Producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los elementos de la Lista de Producto tienen como atributos la descripción, la ordenación, la estimación y el valor.

A menudo, varios Equipos Scrum trabajan juntos en el mismo producto. Para describir el trabajo a realizar sobre el producto, se utiliza una única Lista de Producto. En ese caso podría emplearse un atributo de la Lista de Producto para agrupar varios elementos.

La Lista de Producto puede ser actualizada en cualquier momento por el *Product Owner* pero también el Equipo Scrum puede hacer un refinamiento (*refinement*) de la Lista de Producto, esto es añadir detalles, estimaciones y orden a los elementos de la lista. Se trata de un proceso continuo, en el cual el *Product Owner* y el Equipo de Desarrollo colaboran acerca de los detalles de los elementos de la Lista de Producto. Durante el refinamiento de la Lista de Producto, se examinan y revisan sus elementos.

La Lista de Producto está ordenada, los elementos de orden más alto son más claros y detallados que los de menor orden. Se realizan estimaciones más precisas basándose en la mayor claridad y detalle; cuanto más bajo es el orden, menor es el detalle. Los elementos de la Lista de Producto de los que se ocupará el Equipo de Desarrollo en el siguiente *Sprint* tienen una granularidad mayor, habiendo sido descompuestos de forma que cualquier elemento puede ser “Terminado” dentro de los límites del bloque de tiempo del *Sprint*. Los elementos de la Lista de Producto que pueden ser “Terminados” por el Equipo de Desarrollo en un *Sprint* son considerados “preparados” o “accionables” para ser seleccionados en una Reunión de Planificación de *Sprint*. Los elementos de la Lista de Producto normalmente adquieren este grado de transparencia mediante las actividades de refinamiento descritas anteriormente.

El Equipo de Desarrollo es el responsable de proporcionar todas las estimaciones. El *Product Owner* podría influenciar al Equipo ayudándoles a entender y seleccionar soluciones de compromiso, pero las personas que harán el trabajo son las que hacen la estimación final.

3.3.2. Lista de Pendientes del *Sprint* (*Sprint Backlog*)

Los elementos de la Lista de Producto seleccionados para un *Sprint*, más el plan para terminarlos y entregar el Incremento de producto, y conseguir el Objetivo del *Sprint*, recibe el nombre de Lista de Pendientes del *Sprint* (*Sprint Backlog*). Esta lista la realiza el Equipo de Desarrollo y es éste el único que puede modificarla.

La Lista de Pendientes del *Sprint* es un plan con un nivel de detalle suficiente como para que los cambios en el progreso se puedan entender en la Reunión de Scrum Diario. Durante el *Sprint* el Equipo de Desarrollo podrá modificar la Lista de Pendientes del *Sprint* a medida que trabaja sobre el plan y aprende más acerca del trabajo necesario para conseguir el Objetivo del *Sprint*.

A medida que el trabajo se ejecuta o se completa, se va actualizando la estimación de trabajo restante. Si se requiere nuevo trabajo se añadirá a la lista, así mismo, si algún trabajo ya no es necesario puede ser eliminado. La Lista de Pendientes del *Sprint* es una imagen visible en tiempo real del trabajo que el Equipo de Desarrollo planea llevar a cabo durante el *Sprint*.

3.3.3. Objetivo del *Sprint* (*Sprint Goal*)

El Objetivo del *Sprint* es la meta definida para el *Sprint*, proporciona una guía al Equipo de Desarrollo acerca de por qué se está construyendo el Incremento. Es creado durante la Reunión de Planificación del *Sprint*.

El Objetivo del *Sprint* puede representar otro nexo de unión que haga que el Equipo de Desarrollo trabaje en conjunto y no en iniciativas separadas. A medida que el Equipo de Desarrollo trabaja, se mantiene el Objetivo del *Sprint* en mente. Con el fin de satisfacer el Objetivo del *Sprint* se implementa la funcionalidad y la tecnología. Si el trabajo resulta ser diferente de lo que el Equipo de Desarrollo espera, ellos colaboran con el *Product Owner* para negociar el alcance de la Lista de Pendientes del *Sprint*.

3.3.4. Incremento

El Incremento es la suma de todos los elementos de la Lista de Producto completados durante un *Sprint* y el valor de los Incrementos de todos los *Sprints* anteriores. Al final de un *Sprint*, el nuevo Incremento debe estar "Terminado", lo cual significa que está en condiciones de ser utilizado y que cumple la Definición de "Terminado" del Equipo Scrum. El Incremento debe poder utilizarse aunque el *Product Owner* decida liberarlo o no.

3.3.5. Terminado

Es la definición que se utiliza para evaluar cuándo se ha completado el trabajo sobre el Incremento de producto. Cada Equipo Scrum podrá tener su propia definición de "Terminado" pero debe ser compartida por todo el equipo para asegurar la transparencia. Si hay varios Equipos Scrum trabajando en la entrega del sistema o producto, los Equipos de Desarrolladores en todos los Equipos Scrum deben establecer en conjunto la definición de "Terminado".

Un elemento de la Lista de Producto o un Incremento se describirá como "Terminado" cuando cumpla los requisitos de completado. Con la definición de "Terminado", el Equipo de Desarrollo puede identificar los elementos de la Lista de Producto que seleccionará para la Reunión de Planificación del *Sprint*.

A medida que los Equipos Scrum maduran, se espera que su definición de "Terminado" se amplíe para incluir criterios más rigurosos para una mayor calidad. Cualquier producto o sistema debería tener una definición de "Terminado" que es un estándar para cualquier trabajo realizado sobre él.

3.4. Eventos

Los eventos de Scrum son espacios de tiempo (con una duración máxima) definidos para crear regularidad y ahorrar tiempo de trabajo evitando reuniones innecesarias.

La falta de algunos de estos eventos reduciría la transparencia y podría constituir una oportunidad perdida para la inspección y adaptación.

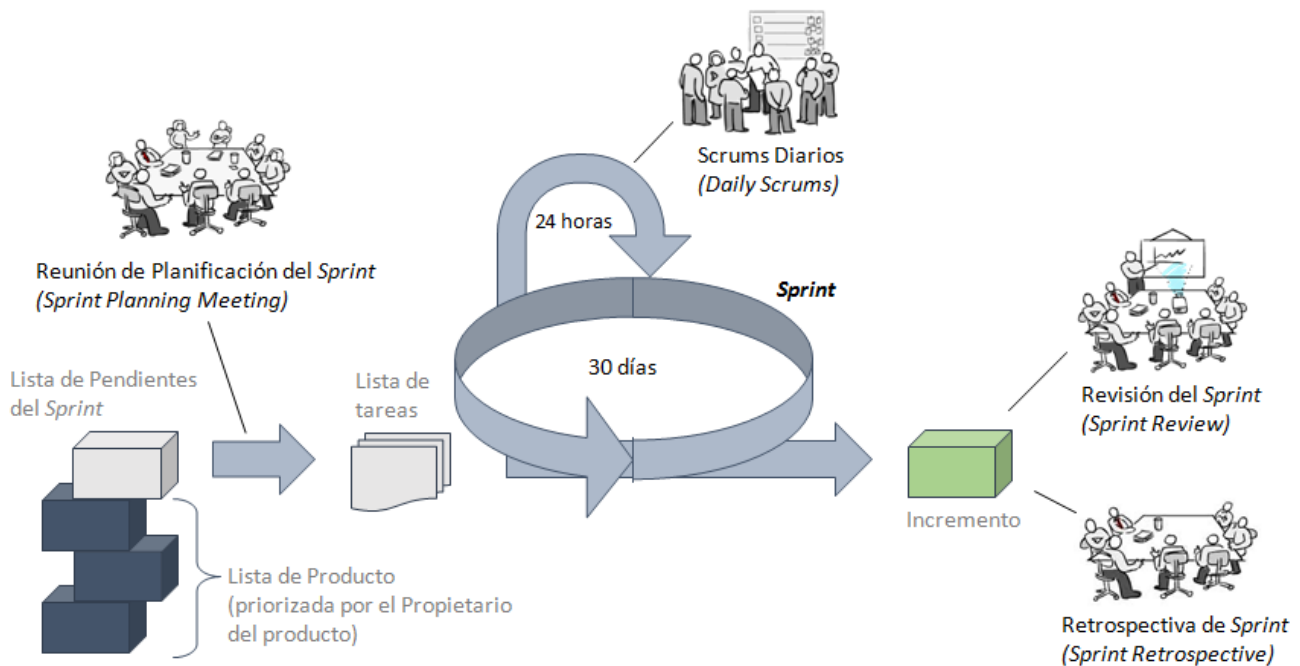


Figura 7: Eventos en el flujo de Scrum

3.4.1. Sprint

Es el espacio de tiempo en el que se desarrolla un Incremento del producto "Terminado", utilizable y potencialmente desplegable; su duración es de un mes o menos, y cada nuevo *Sprint* comienza al finalizar el anterior. Lo más conveniente es que la duración de los *Sprints* sea constante a lo largo del desarrollo.

Los *Sprints* contienen y consisten de la Reunión de Planificación del *Sprint* (*Sprint Planning Meeting*), los Scrums Diarios (*Daily Scrums*), el trabajo de desarrollo, la Revisión del *Sprint* (*Sprint Review*), y la Retrospectiva del *Sprint* (*Sprint Retrospective*).

Durante el *Sprint*:

- No se realizan cambios que puedan afectar al Objetivo del *Sprint*;
- Los objetivos de calidad no disminuyen; y,

- El alcance puede ser clarificado y renegociado entre el *Product Owner* y el Equipo de Desarrollo a medida que se va aprendiendo más.

La duración de los *Sprints* está limitada a un mes ya que si fuera mayor los requerimientos de lo que se está desarrollando podrían cambiar, elevarse la complejidad y aumentar el riesgo. Los *Sprints* habilitan la predictibilidad al asegurar la inspección y adaptación del progreso al menos en cada mes.

3.4.2. Reunión de Planificación del *Sprint* (*Sprint Planning Meeting*)

Es la reunión donde se planifica el trabajo a realizar durante el *Sprint* decidiendo qué se entregará. En ella participa todo el Equipo Scrum: *Product Owner*, Equipo de Desarrollo y el *Scrum Master*. La duración de la reunión será de ocho horas como máximo para un *Sprint* de un mes. Para *Sprints* más cortos, el evento es usualmente más corto.

El *Scrum Master* se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito, éste guiará al Equipo Scrum a mantenerse dentro del bloque de tiempo.

El *Product Owner* discute el objetivo que el *Sprint* debería lograr y los elementos de la Lista de Producto que, si se completan en el *Sprint*, lograrían el Objetivo del *Sprint*.

La entrada a esta reunión está constituida por la Lista de Producto, el último Incremento de producto, la capacidad proyectada del Equipo de Desarrollo para el *Sprint*, y el rendimiento pasado del Equipo de Desarrollo. El número de elementos de la Lista de Producto seleccionados para el *Sprint* depende únicamente del Equipo de Desarrollo. Sólo el Equipo de Desarrollo puede evaluar qué es capaz de lograr durante el *Sprint* que comienza. Después de decidir qué elementos de la Lista de Producto entregará en el *Sprint*, el Equipo Scrum elabora un Objetivo del *Sprint*.

El Equipo de Desarrollo realiza el diseño del sistema y decide cuál es el trabajo necesario para convertir la Lista de Producto en un Incremento de producto funcional. Al final de esta reunión, el trabajo planificado por el Equipo de Desarrollo para los primeros días del *Sprint* es descompuesto en unidades de un día o menos. El Equipo de Desarrollo se autoorganiza para asumir el trabajo de la Lista de Pendientes del *Sprint*, tanto durante la reunión de Planificación del *Sprint* como a lo largo del *Sprint*.

El *Product Owner* puede ayudar a clarificar los elementos de la Lista de Pendientes del *Sprint* y hacer concesiones. Si el Equipo de Desarrollo determina que tiene demasiado trabajo o que no tiene suficiente trabajo, podría renegociar los elementos de la Lista de Pendientes del *Sprint* con el *Product Owner*. El Equipo de Desarrollo podría también invitar a otras personas a que asistan con el fin de que proporcionen asesoría técnica o relacionada con el dominio.

3.4.3. Scrums Diarios (*Daily Scrums*)

El Scrum Diario es una reunión de unos 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente.

El Scrum Diario se realiza a la misma hora y en el mismo lugar todos los días para reducir la complejidad. Durante la reunión, cada miembro del Equipo de Desarrollo explica:

- ¿Qué hice ayer que ayudó al Equipo de Desarrollo a lograr el Objetivo del *Sprint*?
- ¿Qué haré hoy para ayudar al Equipo de Desarrollo a lograr el Objetivo del *Sprint*?
- ¿Veo algún impedimento que evite que el Equipo de Desarrollo o yo logremos el Objetivo del *Sprint*?

En esta reunión el Equipo de Desarrollo evalúa el progreso hacia el Objetivo del *Sprint* y qué tendencia sigue este progreso hacia la finalización del trabajo contenido en la Lista de Pendientes del *Sprint*. El Equipo de Desarrollo o los miembros del equipo a menudo se vuelven a reunir inmediatamente después del Scrum Diario, para tener discusiones detalladas, o para adaptar, o replanificar el resto del trabajo del *Sprint*.

El *Scrum Master* se asegura de que el Equipo de Desarrollo tenga la reunión, pero el Equipo de Desarrollo es el responsable de dirigir el Scrum Diario. El *Scrum Master* guía al Equipo de Desarrollo para que la duración de la reunión sea de 15 minutos, además se asegura de que se cumpla la regla de que sólo los miembros del Equipo de Desarrollo participan en el Scrum Diario.

Los Scrum Diarios mejoran la comunicación, eliminan la necesidad de mantener otras reuniones, identifican y eliminan impedimentos relativos al desarrollo, resaltan y promueven la toma de decisiones rápida, y mejoran el nivel de conocimiento del Equipo de Desarrollo. El Scrum Diario constituye una reunión clave de inspección y adaptación.

3.4.4. Revisión del *Sprint* (*Sprint Review*)

Al final del *Sprint* se lleva a cabo una Revisión del *Sprint* para inspeccionar el Incremento y adaptar la Lista de Producto si fuese necesario. Durante la Revisión del *Sprint*, el Equipo Scrum y los interesados conversan acerca de lo que se hizo durante el *Sprint*. Basándose en esto, y en cualquier cambio de la Lista de Producto durante el *Sprint*, los asistentes determinan las siguientes cosas que podrían hacerse para optimizar el valor. Se trata de una reunión informal, no una reunión de seguimiento, y la presentación del Incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración.

La duración de esta reunión para un *Sprint* de un mes es de cuatro horas. Como en la Reunión de Planificación del *Sprint*, el *Scrum Master* se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito, éste guiará al Equipo Scrum a mantenerse dentro del bloque de tiempo.

La Revisión del *Sprint* incluye los siguientes elementos:

- Los asistentes son el Equipo Scrum y los interesados clave invitados por el *Product Owner*;
- El *Product Owner* explica qué elementos de la Lista de Producto se han “Terminado” y cuáles están pendientes proyectando fechas de finalización probables en el tiempo basándose en el progreso obtenido hasta la fecha (si es necesario);
- El Equipo de Desarrollo habla acerca de qué fue bien durante el *Sprint*, qué problemas aparecieron y cómo fueron resueltos esos problemas;
- El Equipo de Desarrollo demuestra el trabajo que ha “Terminado” y responde preguntas acerca del Incremento;
- El grupo completo colabora acerca de qué hacer a continuación, de modo que la Revisión del *Sprint* proporcione información de entrada valiosa para las Reuniones de Planificación de *Sprints* subsiguientes;
- Se revisa cómo el mercado o el uso potencial del producto podría haber cambiado lo que es de más valor para hacer a continuación; y,
- Se estudia la línea de tiempo, presupuesto, capacidades potenciales y mercado para la próxima entrega prevista del producto.

El resultado de la Revisión del *Sprint* es una Lista de Producto revisada, que define los elementos de la Lista de Producto posibles para el siguiente *Sprint*. Es posible además que la Lista de Producto reciba un ajuste general para enfocarse en nuevas oportunidades.

3.4.5. Retrospectiva del *Sprint* (*Sprint Retrospective*)

La Retrospectiva de *Sprint* es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente *Sprint*. Después de la Revisión del *Sprint* y antes de la siguiente Reunión de Planificación del *Sprint* se lleva a cabo la Retrospectiva del *Sprint*.

Esta reunión tiene una duración máxima de tres horas para *Sprints* de un mes, el *Scrum Master* se asegurará de que la reunión se realice cumpliendo los horarios y los asistentes entiendan su propósito.

El propósito de la Retrospectiva del *Sprint* es:

- Inspeccionar cómo fue el último *Sprint* en cuanto a personas, relaciones, procesos y herramientas;

- Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras; y,
- Crear un plan para implementar las mejoras a la forma en la que el Equipo Scrum desempeña su trabajo.

Durante la Retrospectiva del *Sprint*, el Equipo Scrum planifica formas de aumentar la calidad del producto mediante la adaptación de la definición de "Terminado". De esta forma al finalizar la reunión, el Equipo Scrum habrá identificado mejoras que implementará en el próximo *Sprint*.

Aunque las mejoras pueden implementarse en cualquier momento, la Retrospectiva del *Sprint* ofrece un evento dedicado para este fin, enfocado en la inspección y la adaptación.

3.5. Otras Herramientas

Las herramientas que se pueden utilizar en las metodologías ágiles ayudan a que la gestión de proyectos y aplicación de los métodos sean más eficientes. La colaboración de todo el equipo es fundamental para el éxito de las mismas.

3.5.1. *Planning Poker*

Planning Poker es una técnica ideada por James Grenning para la estimación inicial del proyecto, con ella se evalúa el esfuerzo y la duración de las tareas.

El planteamiento se basa en la Lista de Historias de Usuario que recogen la descripción de los requerimientos del usuario, y una baraja de cartas que tendrá cada miembro del equipo como la que se muestra en la imagen:



Figura 8: Baraja de cartas de *Planning Poker* [12]

Cada equipo u organización puede utilizar un juego de cartas con las numeraciones adecuadas a la unidad de esfuerzo con la que trabajan, y el tamaño máximo de tarea que se va a estimar. La baraja mostrada en la figura utiliza la sucesión de Fibonacci más dos cartas especiales: interrogación (indica que por alguna razón no se puede precisar la estimación) y *cocktail* (indica que se necesita un descanso).

Cada número significa un peso / esfuerzo / complejidad para completar un objetivo (Historia de Usuario). Según aumenta la numeración de las cartas, los valores de éstas se van haciendo mayor y se van distanciando para que, de esta forma, se facilite la decisión sobre qué tamaño tiene un objetivo (cuanto mayor sea una estimación, mayor es la incertidumbre). Así, por ejemplo, en el caso de usar la baraja de la figura, una persona cree que el tamaño adecuado de una tarea es 6, se ve obligado a reflexionar y elegir el 5 aceptando que parte de la incertidumbre percibida no existe, o bien escoger el 8 y optar por una estimación más conservadora de la incertidumbre.

La operativa es la siguiente:

En la reunión de estimación el *Product Owner* lee una Historia de Usuario y el equipo hace las preguntas que considere para entender bien la característica propuesta.

Cada miembro del equipo piensa en el esfuerzo necesario para completar el objetivo y todos muestran sus tarjetas simultáneamente, de esta manera nadie está condicionado por las estimaciones de los otros. Las personas que han sacado una carta con un valor mucho más alto o bajo explican por qué su votación es tan diferente al resto y se debate, ya que es posible que conozcan algún problema que el resto no haya considerado o por el contrario sepan cómo resolver de manera sencilla un problema.

Se vuelve a votar hasta que se llega a un consenso, si no se llega a un acuerdo en un tiempo determinado se puede:

- Dejar a un lado la estimación de esa tarea y retomar al final o en otro momento aquellas que hayan quedado pendientes.
- Pedir al cliente o *Product Owner* que descomponga la funcionalidad y valorar cada una de las funcionalidades resultantes.
- Tomar la estimación menor, mayor, o la media.

Con esta técnica se evita que el equipo se estanque y se alargue el tiempo de la reunión, y consigue además que participe todo el equipo expresando su opinión y aportando su conocimiento.

3.5.2. Pizarra de Tareas (*Scrum Taskboard*)

La pizarra de tareas de Scrum muestra el estado del flujo de trabajo del proyecto, su objetivo principal es gestionar de manera general como se van completando tareas.

Se basa en una pizarra dividida en columnas las cuales representan el estado por el que una tarea puede pasar. Cada equipo de trabajo puede crear su propia pizarra y organizarla como más le convenga. Una de las pizarras típicas es aquella que consta de tres columnas: Pendiente, En progreso y Terminado.

Se utilizan tarjetas (generalmente post-it para poder mover por el tablero) donde se escriben las tareas a realizar. Inicialmente estas tarjetas se colocan en la primera columna al lado de cada Historia de Usuario u objetivo al que pertenezcan y se irán moviendo hacia la derecha para cambiarlas de estado; para cada miembro del equipo se pueden utilizar adhesivos de colores más pequeños sobre cada tarea, de manera que se pueda ver en qué tareas está trabajando cada cual. En la Figura 9 se puede ver un ejemplo de *Taskboard*.



Figura 9: *Scrum Taskboard* [13]

Cada integrante del equipo se responsabilizará de una tarea e irá desplazando la tarjeta correspondiente según se desarrolle, así la pizarra estará actualizada al instante y el seguimiento de la misma será en tiempo real. De un simple vistazo se podrá observar cuántas tareas están pendientes de realizar, cuántas en progreso y cuántas finalizadas. El movimiento de tarjetas provoca un efecto motivador al equipo.

Las pizarras de tareas favorecen la comunicación directa del equipo al actualizar la información en reuniones frente a ellas además de compartir la visibilidad de la evolución del proyecto con todos los implicados.

La exposición de las tareas en estas pizarras facilita la detección temprana de problemas al monitorizar continuamente la evolución del proyecto. La actualización de la información *just-in-time*, ayuda a identificar en un primer momento los posibles impedimentos, problemas y riesgos, que de otra forma pasan desapercibidos hasta que empiezan a producir retrasos o repercusiones ya inevitables.

3.5.3. Gráfica *Burn Down*

La gráfica *Burn Down* representa el trabajo pendiente de realizar a lo largo del tiempo, es decir, la velocidad a la que se están completando los objetivos.

El Equipo de Desarrollo actualiza este gráfico en las Reuniones de Revisión del *Sprint* consiguiendo monitorizar el ritmo del avance así como detectar de forma temprana posibles desviaciones sobre la previsión que pudieran comprometer la entrega al final del *Sprint*.

El equipo dispone en la Lista de Pendientes del *Sprint* las tareas que se deben realizar y el esfuerzo pendiente que se ha estimado para cada una. Día a día cada miembro del equipo actualiza en la Lista de Pendientes del *Sprint* el tiempo que le queda a las tareas que va desarrollando, hasta que se terminan quedando a cero el tiempo pendiente. Con esta información se actualiza el gráfico poniendo cada día el esfuerzo pendiente total de todas las tareas que aún no se han terminado (en el eje Y del gráfico se registra el trabajo que aún falta por realizar y en el eje X se detallan los días del *Sprint*).

El avance ideal de un *Sprint* estaría representado por la diagonal que reduce el esfuerzo pendiente de forma continua y gradual hasta terminarlo en el último día del *Sprint* tal y como se muestra en la Figura 10.

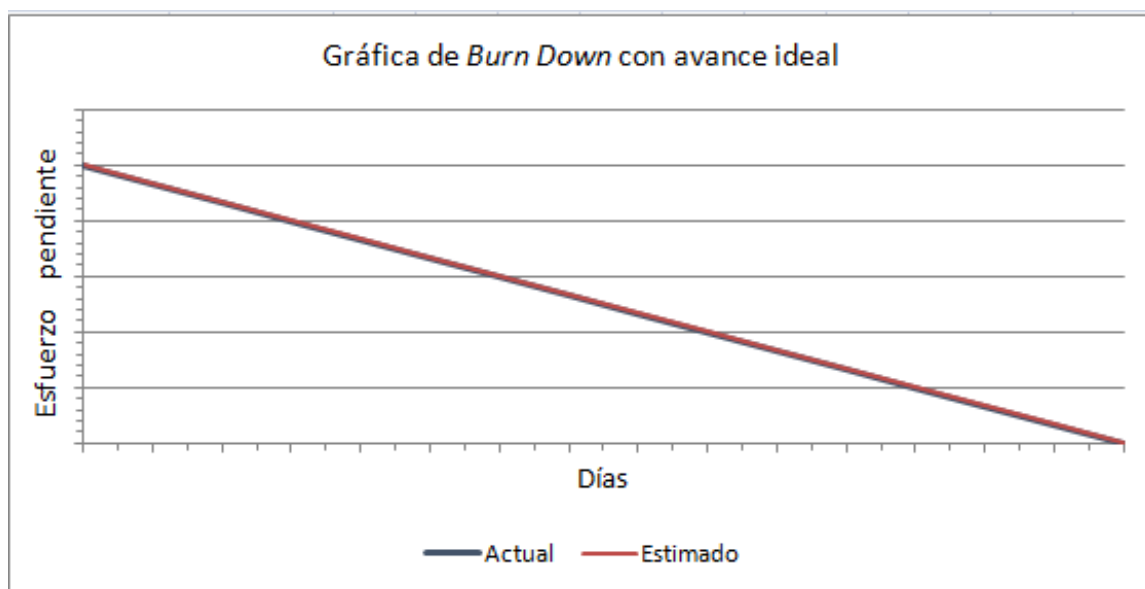


Figura 10: Gráfica de *Burn Down* con avance ideal

Las gráficas con avance ideal no son habituales y normalmente encontramos gráficas como la representada en la Figura 11:

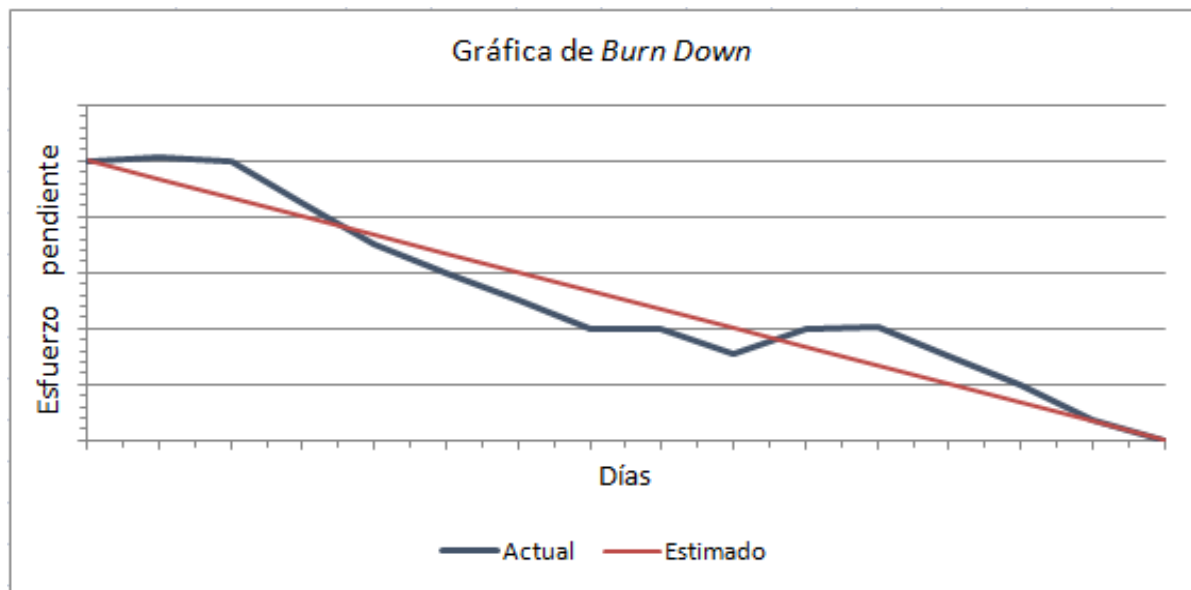


Figura 11: Gráfica de *Burn Down* habitual

Quando la estimación del esfuerzo realizada en la Reunión de Planificación es mayor que el esfuerzo real que implican las tareas, se dice que la gráfica sigue un patrón de “*Sprint Subestimado*” como se muestra en la Figura 12:

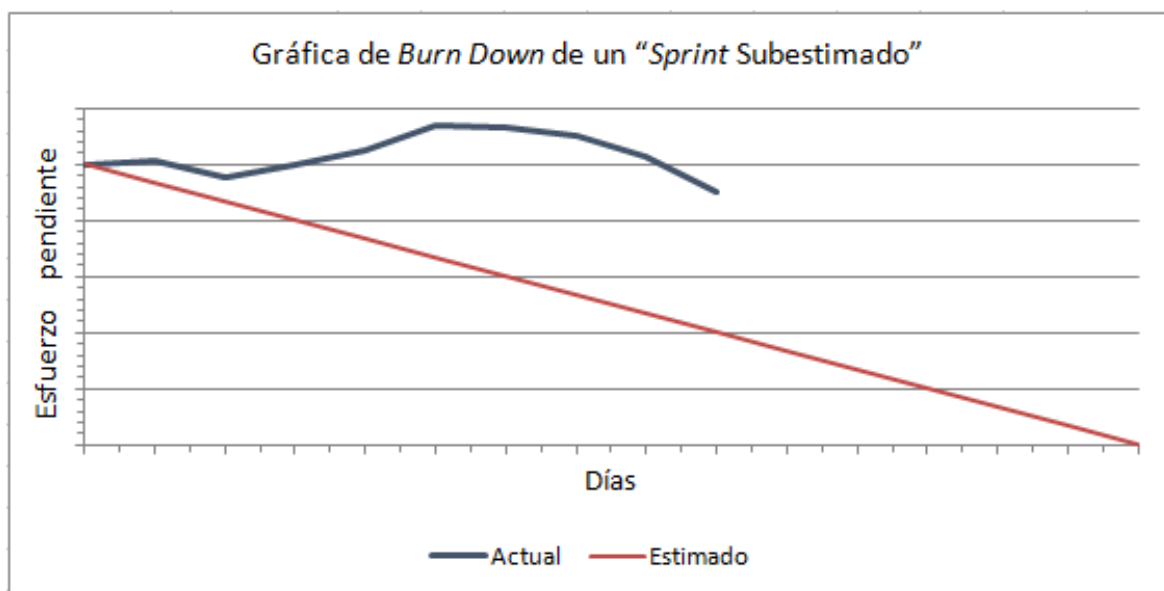


Figura 12: Gráfico de *Burn Down* de un “*Sprint Subestimado*”

En cambio, cuando el esfuerzo estimado es menor al real, se dice que la gráfica sigue un patrón de “*Sprint Sobrestimado*” como se muestra en la Figura 13.

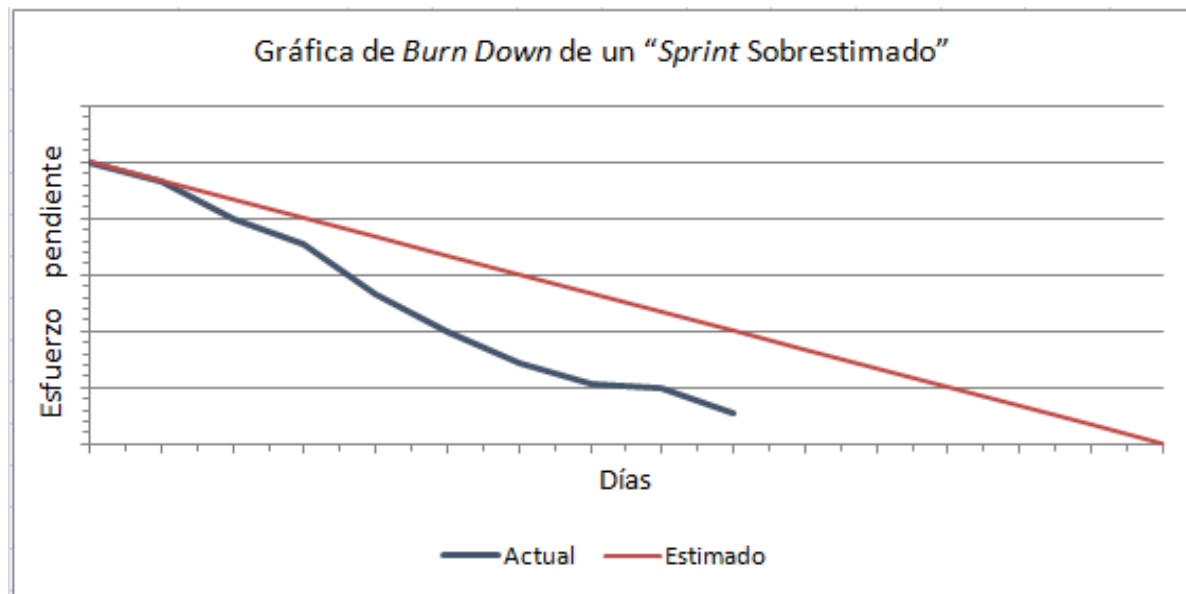


Figura 13: Gráfico de *Burn Down* de un “*Sprint Sobrestimado*”

3.6. El proceso de Scrum

Una vez analizados los diferentes elementos que constituyen Scrum, veremos cómo aparecen en escena y se relacionan entre ellos durante el desarrollo del proceso Scrum. De esta manera conseguiremos una visión global y precisa del mismo.

En Scrum un proyecto se desarrolla en iteraciones en las cuales se proporciona un resultado completo, es decir, un Incremento de producto final que sea susceptible de ser entregado al receptor del proyecto.

En primer lugar, durante la primera parte de la Reunión de Planificación del *Sprint* (4 horas máx.), se realiza de forma conjunta con el cliente la toma de requisitos del producto mediante una definición sencilla y clara de las características que debe tener el producto, definiendo las Historias de Usuario que van a guiar el proceso de desarrollo.

Las Historias de Usuario son la representación de las características que el usuario espera que tenga su producto, éstas se escriben utilizando un lenguaje sencillo y cercano al usuario de forma que sea comprensible por los clientes y los desarrolladores. Las Historias de Usuario son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos.

Con las Historias de Usuario se realiza una primera versión de la Lista de Producto donde el cliente ha priorizado los objetivos balanceando el valor que le aportan respecto a su coste, éstos quedan repartidos en iteraciones y entregas que se realizarán en etapas de corta duración denominadas *Sprints*. De manera regular el cliente puede maximizar la utilidad de lo que se desarrolla y el retorno de inversión mediante la replanificación de objetivos del producto con vista a las siguientes iteraciones.

El Equipo Scrum examina la lista y pregunta al cliente las dudas que le surgen, entonces selecciona los objetivos/requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

En la segunda parte de la Reunión de Planificación del *Sprint* (4 horas máx.), el Equipo de Desarrollo elabora la Lista de Pendientes del *Sprint* definiendo las tareas necesarias para desarrollar los requisitos a los que se ha comprometido. Se realiza la estimación de esfuerzo de forma conjunta y cada miembro del equipo se autoasigna las tareas que puede realizar. Esta reunión está supervisada por el *Product Owner* que asegurará que aquello que es más prioritario sea lo primero que se realice.

Tras la Reunión de Planificación del *Sprint* comienza el *Sprint* (4 semanas máx.), durante este tiempo el equipo transformará la Lista de Pendientes del *Sprint* en un producto funcional y listo para entregar, pasando las fases de análisis, diseño, desarrollo, test y pruebas para todas las Historias que haya que implementar. En el trabajo diario cada miembro del equipo toma aquellas tareas que considera oportunas en cada momento, para ello se hace uso de paneles de seguimiento donde se tiene una fácil visión gráfica del estado del *Sprint* en su totalidad y sus partes.

Se hará una reunión diaria, *Scrum Diario* (15 minutos máx.), en la que cada miembro del equipo expone lo que ha hecho y lo que realizará hasta la siguiente reunión, y si está teniendo algún problema, o si prevé que pueda encontrar algún impedimento. Es una reunión operativa, informal y ágil con el Equipo de Desarrollo en la que también se actualiza el gráfico de avance del *Sprint* con las estimaciones actualizadas.

Una vez finalizado el *Sprint* se obtiene un Incremento, parte del producto, un entregable o algo que se pueda mostrar y que enseñe los avances acometidos en el *Sprint*. En este punto se procede a la Reunión de Revisión del *Sprint* (4 horas máx.), esta reunión marca, a intervalos regulares, el ritmo de construcción, y la trayectoria que va tomando la visión del producto. Es una reunión informativa en la que se demuestra el funcionamiento de las partes construidas determinando qué elementos de la Lista de Producto están en estado "Terminado" y cuáles no, se revisa la Lista de Producto pudiendo añadir nuevas Historias de Usuario o replantear las ya existentes, y se convoca la próxima Reunión de Planificación del *Sprint*.

Para mantener la calidad del producto se establece que una Historia de usuario está 100% completa si supera los test unitarios, pasa las pruebas de aceptación, supera los test adicionales para mantener la calidad del producto, el producto está construido e integrado satisfactoriamente en el entorno del cliente y, por supuesto, es aceptado por éste. Todo aquello que se presente se considera "Terminado", por tanto, si alguna de las Historias previstas no se termina cuando lo hace el *Sprint*, ésta no se presenta y se considera un fracaso a nivel de planificación.

Tras la realización de cada *Sprint* también se lleva a cabo una Reunión de Retrospectiva del *Sprint* que permite aprender de los conocimientos y experiencias adquiridas hasta el momento y mejorar de

forma continua el proceso de desarrollo. No se debe confundir con la Reunión de Revisión del *Sprint* ya que ésta analiza qué se está construyendo, mientras que la Reunión de Retrospectiva del *Sprint* se centra en cómo se está construyendo. Se revisará con el equipo los objetivos marcados inicialmente en la Lista de Producto concluida, se aplicarán los cambios y ajustes si son necesarios, y se marcarán los aspectos positivos (para repetirlos) y los aspectos negativos (para evitar que se repitan) del *Sprint*, que servirán de retroalimentación para el siguiente *Sprint*.

Posteriormente se hace una reunión para planificar el siguiente *Sprint*, repitiendo todo el proceso hasta que la Lista de Producto se termina o bien el proyecto se dé por terminado.

4. CASO DE ESTUDIO

A continuación se detalla el caso de estudio elegido para este proyecto a fin de entender la metodología *Scrum* y su implantación.

El objetivo del caso de estudio consistirá en definir un Servicio Gestionado basado en el modelo iterativo *Scrum* para un cliente del Sector Público que desea externalizar su Servicio de Orientación para la Búsqueda de Empleo y Creación de Puestos de Trabajo. Por tanto, este servicio tendrá como objetivo determinar cuáles son los servicios informáticos necesarios para desarrollar y mantener los sistemas informáticos y aplicaciones que garanticen la gestión de recursos y actividades que permitan al Cliente prestar su servicio a los ciudadanos y empresas.

Se detallarán los trabajos a realizar así como los recursos humanos que se estiman necesarios para el objetivo de este trabajo.

Para comprender bien la interrelación de cada uno de los elementos que conforman el marco de referencia *Scrum*, detallaremos el proyecto global desde los diferentes modelos de trabajo del Servicio Gestionado que se prestará: Operativo, Organizativo, de Gestión y Económico.



Figura 14: Modelos de un Servicio Gestionado

4.1. Modelo Operativo

Inicialmente se realizará una descripción del Modelo Operativo y de Relación del Servicio Gestionado que nos permitirá establecer un servicio de mantenimiento flexible y operativo con parámetros de calidad claros y medibles que nos darán acceso no sólo a controlar su rendimiento sino también a mejorar a través de un proceso de evolución continua.

A continuación se detalla dicho Modelo Operativo:



Figura 15: Modelo Operativo de un Servicio Gestionado

4.1.1. Alcance y Líneas de Servicio

A continuación se describe la visión de servicios que soporta el modelo operativo para dar respuesta a las necesidades del Cliente.

Las actividades comunes proporcionadas por este servicio para todas las áreas que conforman el equipo son las siguientes:

- Mantenimiento de la relación con las diferentes áreas de gestión (apoyándose en otras capas del servicio cuando la operatividad de la relación así lo requiera).
- Gestión de la cartera de trabajos a realizar y los recursos asignados al servicio.
- Elaboración y revisión de los distintos informes que formalizan la comunicación entre los Proveedores y el Cliente para el servicio prestado.

- Monitorización, con el propósito de observar el funcionamiento del servicio y recoger la información necesaria que permita realizar propuestas de mejora del mismo, permitiendo un realineamiento con la evolución del negocio.
- Gestión de la incorporación de nuevos desarrollos.

Las distintas líneas de servicio permiten realizar la ejecución del mismo, adecuándose a la estrategia fijada por la gestión del servicio, y los estándares y controles de calidad establecidos por la gestión de la calidad. Deben proporcionar el conjunto de procedimientos, técnicas y herramientas que posibilitan la ejecución de las peticiones de trabajo solicitadas, consiguiendo los resultados perseguidos desde el punto de vista funcional, técnico, de calidad y de alineamiento con las necesidades del negocio.

Por tanto, a través de este servicio se pretende conseguir la generación de entregables que den una respuesta adecuada a las necesidades del negocio y al modelo de desarrollo y mantenimiento planteado por el cliente, así como el objetivo de mejorar continuamente los procesos soportados y automatizados por las aplicaciones. Se entiende como "entregable" las salidas de cada uno de los procesos que intervienen en el Servicio Gestionado ofrecido tales como documentos, informes, aplicaciones software, actividades, etc.

4.1.1.1. Adaptación Nueva Convocatoria

Se llevarán a cabo las actuaciones que sean necesarias para permitir la gestión de las nuevas Convocatorias de Iniciativas de Orientación que puedan surgir, tomando como base, en la medida que esto sea posible, las funcionalidades desarrolladas para la Convocatoria que se tome como origen de la adaptación.

Las tareas a realizar serán:

- Análisis de viabilidad de la adaptación e impacto sobre la aplicación y la operativa.
- Adaptar los sistemas (programación y/o configuración).
- Realización de los planes de prueba que cubran, al menos, la funcionalidad existente para la Convocatoria de origen.
- Despliegue de las adaptaciones del sistema en los distintos entornos.

4.1.1.2. Mantenimiento Correctivo

Este servicio llevará a cabo la resolución de incidencias funcionales y técnicas detectadas por los usuarios y que requieren la modificación/adaptación de los programas desarrollados o las parametrizaciones realizadas en las aplicaciones.

Las tareas a realizar serán:

- Diagnósticos del error.
- Análisis funcional y técnico de la solución.
- Detección de problemas recurrentes y su consiguiente resolución.
- Mantenimiento de la documentación técnica y funcional.
- Implantar soluciones de contingencia.
- Desarrollar las modificaciones a los sistemas (programación y/o configuración), incluyendo pruebas unitarias documentales.
- Detección de problemas recurrentes, identificación, análisis, diseño y ejecución de correcciones necesarias para la resolución de problemas.
- Actualizar el estado de la incidencia.
- Mantener los Manuales de Uso de las aplicaciones.
- Analizar los problemas de rendimiento en transacciones concretas producidos en el desarrollo o en la configuración.

4.1.1.3. Mantenimiento Preventivo

Esta línea de servicio se encargará de realizar tareas de mejora en los sistemas para proporcionar incrementos de rendimiento o adaptaciones a nuevos entornos hardware o versiones de software.

Las tareas principales que se realizarán son:

- Estudio de las mejoras a realizar en las aplicaciones.
- Análisis de viabilidad e impacto sobre la aplicación y la operativa.
- Estudio de alternativas en la implantación de ciertas especificaciones.
- Propuestas de cambios en las aplicaciones para la mejora de los sistemas.
- Supervisión del rendimiento de las aplicaciones.

4.1.1.4. Mantenimiento Evolutivo

Se llevarán a cabo los desarrollos necesarios que permitan cubrir las nuevas funcionalidades, solicitadas por las unidades usuarias, o modificaciones a las funcionalidades existentes y que permitan agilizar y mejorar su gestión y sus procesos.

Evolutivo menor: Consiste en la implantación, a petición del usuario, de cambios en el sistema debido a pequeñas modificaciones en las especificaciones del mismo y con un bajo tiempo de ejecución.

Las tareas a llevar a cabo en esta línea de servicio son:

- Análisis funcional y técnico de los cambios, mejoras y nuevas funcionalidades.
- Soporte a los análisis de impacto y estudios de viabilidad de los cambios a realizar.
- Desarrollo de las modificaciones solicitadas.
- Pruebas unitarias e integradas.
- Plan de pruebas funcional.
- Mantenimiento de la documentación técnica y funcional.

Evolutivo mayor: Relativo a nuevas funcionalidades que, por su naturaleza y complejidad requieren un alto esfuerzo de desarrollo, excediendo el ámbito del evolutivo menor, y por tanto requieren un tratamiento independiente y deben ser planificadas y acometidas dentro del plan general del proyecto de forma independiente.

Evolutivo legal o normativo: modificaciones derivadas del cambio en la legislación.

4.1.1.5. Nuevos Desarrollos

Se llevarán a cabo los desarrollos necesarios que permitan cubrir las nuevas aplicaciones solicitadas por las unidades usuarias.

Las tareas a realizar serán:

- Análisis funcional, técnico y manual de uso de la solución.
- Desarrollar los sistemas (programación y/o configuración).
- Realización de los planes de prueba que cubran la práctica totalidad de las historias de usuario definidas.
- Despliegue del sistema en los distintos entornos.
- Elaborar manuales de uso de las aplicaciones.

4.1.1.6. Soporte a Usuarios

Este servicio llevará a cabo la resolución de las consultas formuladas por los usuarios.

Las tareas a realizar serán:

- Soporte directo o consultas:
 1. Atención y respuesta a dudas, consultas y peticiones de usuarios sobre las aplicaciones.
 2. Ayuda para la corrección de errores por datos introducidos erróneamente por el usuario.
 3. Registro y solución de consultas a los usuarios de los aplicativos sobre el modo de operación.
 4. Realización de consultas de datos a petición del usuario.

4.1.2. Definición de Proyecto y ciclo de vida

Un proyecto es un "Esfuerzo temporal acometido para crear un único servicio o producto". Existen infinidad de definiciones de Proyecto, pero la anteriormente mencionada es la que propone el PMI (*Project Management Institute*) [15] en su compendio de buenas prácticas orientadas a la gestión de proyectos y se ajusta muy bien a la intención que hay en el cliente de encaminar sus necesidades informáticas de ahora en adelante.

Atendiendo a las fases de gestión de proyectos, se desarrollará el proyecto siguiendo el modelo Scrum, mostrando sus procesos, entregables, y los actores implicados en cada uno de ellos.

Ciclo de Vida de un Proyecto

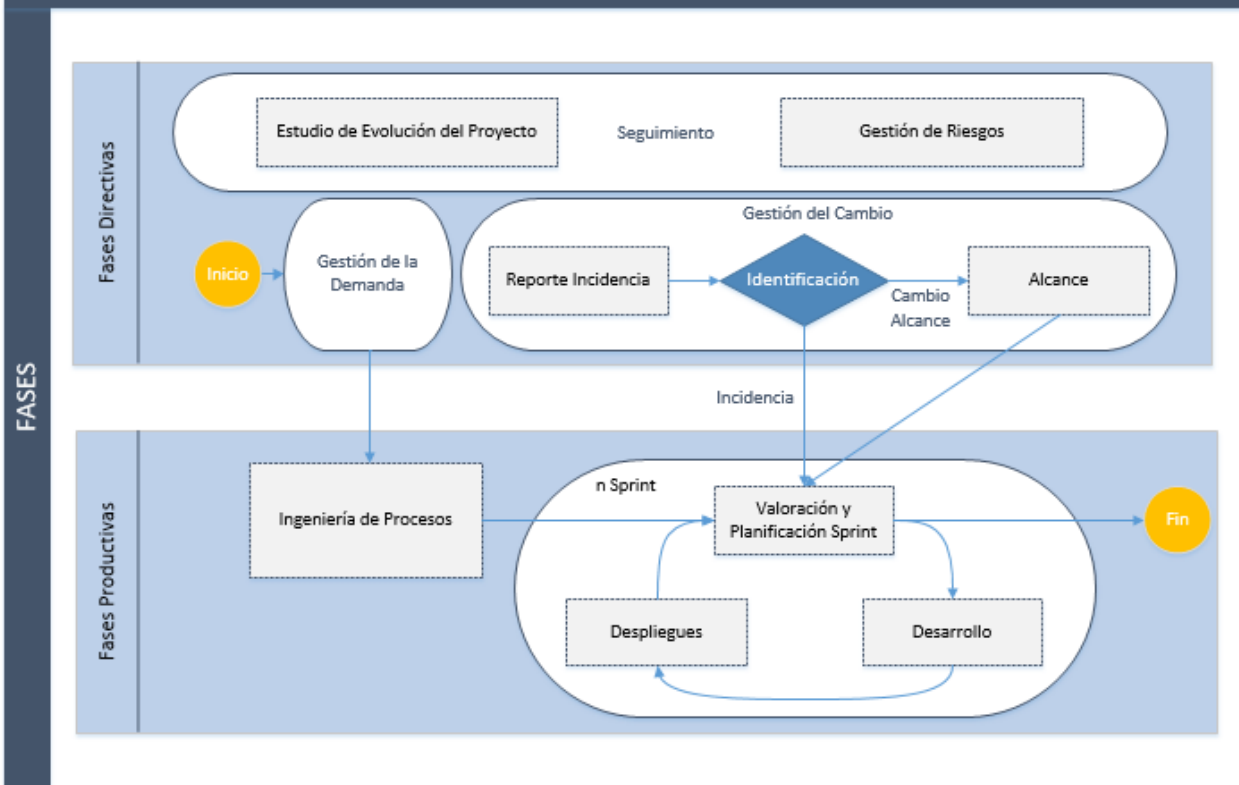


Figura 16: Ciclo de Vida de un Proyecto

A continuación se exponen las fases del ciclo de vida del proyecto que se desarrollan durante todo el ciclo independientemente de la fase secuencial en la que nos encontremos.

4.1.2.1. Fases Directivas

a) Gestión de la Demanda

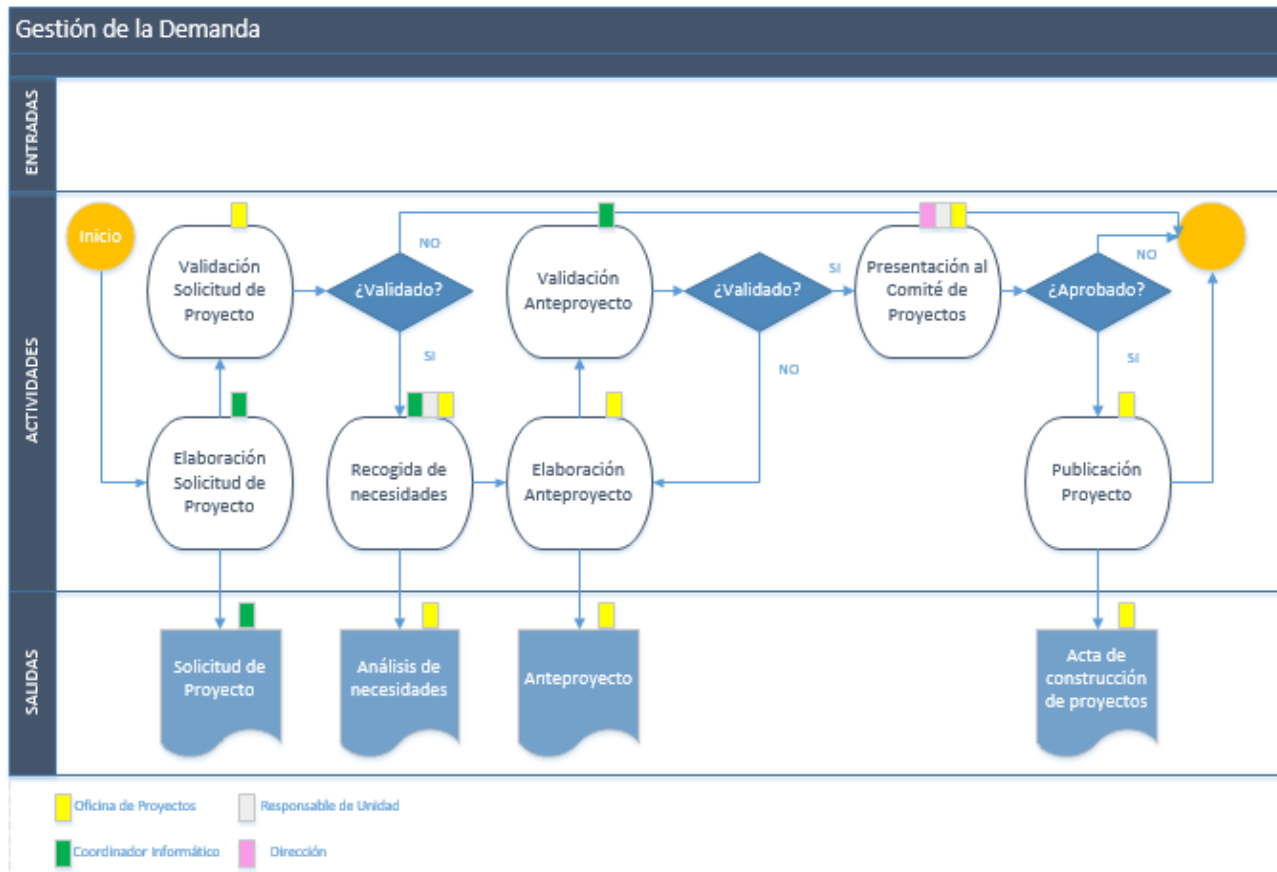


Figura 17: Gestión de la Demanda

Responsable

La responsabilidad de esta fase recae sobre la Oficina de Proyectos, que es la encargada de atender a los usuarios a la hora de plantear sus nuevas necesidades que pudieran acabar en proyecto.

Actores Involucrados

- Oficina de Proyectos
- Coordinador Informático
- Responsable de la Unidad de Proyectos Informáticos
- Responsable de la Unidad de Servicios Informáticos
- Dirección de Sistemas Información

Entradas

No es necesaria ninguna entrada

Actividades

1. El Coordinador Informático recopila las necesidades de desarrollo que surgen en el equipo

<p>de usuarios de las iniciativas y aplicativos del cliente, y las presenta a la Oficina de Proyectos como una Solicitud de Proyecto.</p>	
2.	La Oficina de Proyectos se encarga de validar la Solicitud de Proyecto presentada y comprobar que la información aportada pueda iniciar la tramitación de un nuevo proyecto.
3.	El Coordinador Informático se pone en contacto con la Oficina de Proyectos para exponer sus necesidades con el fin de realizar un análisis más detallado del proyecto. Se convoca una reunión con el Coordinador Informático, la Oficina de Proyectos y el Responsable de la Unidad, donde se plasma dicha necesidad.
4.	La Oficina de Proyectos genera un anteproyecto con toda la información obtenida del usuario para poder exponerla y defenderla de manera imparcial ante el Comité de Proyectos.
5.	El Coordinador Informático revisa en anteproyecto con el fin de validar el mismo. En caso de ser necesario se corrigen todos los puntos que se consideren oportunos.
6.	Se reúne el Comité de Proyectos, formado por la Dirección de Sistemas de Información, los Responsables de las Unidades de Proyectos y los Servicios Informáticos y el Responsable de la Oficina de Proyectos. El Responsable de la Oficina de Proyectos presenta el anteproyecto y se valora para decidir si se convierte en Proyecto. En la toma de decisión no interviene el Responsable de la Oficina de Proyectos. En función de la decisión tenemos tres opciones:
a)	Se aprueba el anteproyecto convirtiéndose en proyecto y se asigna un <i>Product Owner</i> que gestionará el desarrollo del proyecto.
b)	Se desestima el anteproyecto y finaliza el proceso.
c)	Se pospone la decisión al no tener suficiente información. Se solicita la misma al Coordinador Informático para poder volver a tratar el anteproyecto en el siguiente comité de proyectos. Se vuelve al proceso de Recogida de necesidades y se continúa con el flujo establecido.
Salidas	
<p>Al finalizar el proceso de la demanda, dependiendo del avance en el flujo, se obtiene:</p> <ul style="list-style-type: none"> • Una Solicitud de Proyecto, con la exposición por parte del Coordinador de las necesidades iniciales expuestas. • Documento de Análisis de las Necesidades, con las necesidades expuestas a lo largo de la fase de Recogida de Necesidades en las reuniones y desglosadas en detalle. • Anteproyecto, con la descripción del proyecto y análisis de la viabilidad. • Acta de constitución de Proyecto, con los objetivos, alcance y responsable de la gestión del proyecto. 	

Tabla 2: Gestión de la Demanda

b) Gestión del Cambio

En la gestión del Cambio se trata tanto el Reporte de Incidencias como los Cambios de Alcance.

i. Reporte de Incidencias

Se considera incidencia cualquier error que se produzca en la aplicación, que impida el normal funcionamiento de la misma, y que deriva de una mala construcción, y no se detecta hasta que no está funcionando en producción.

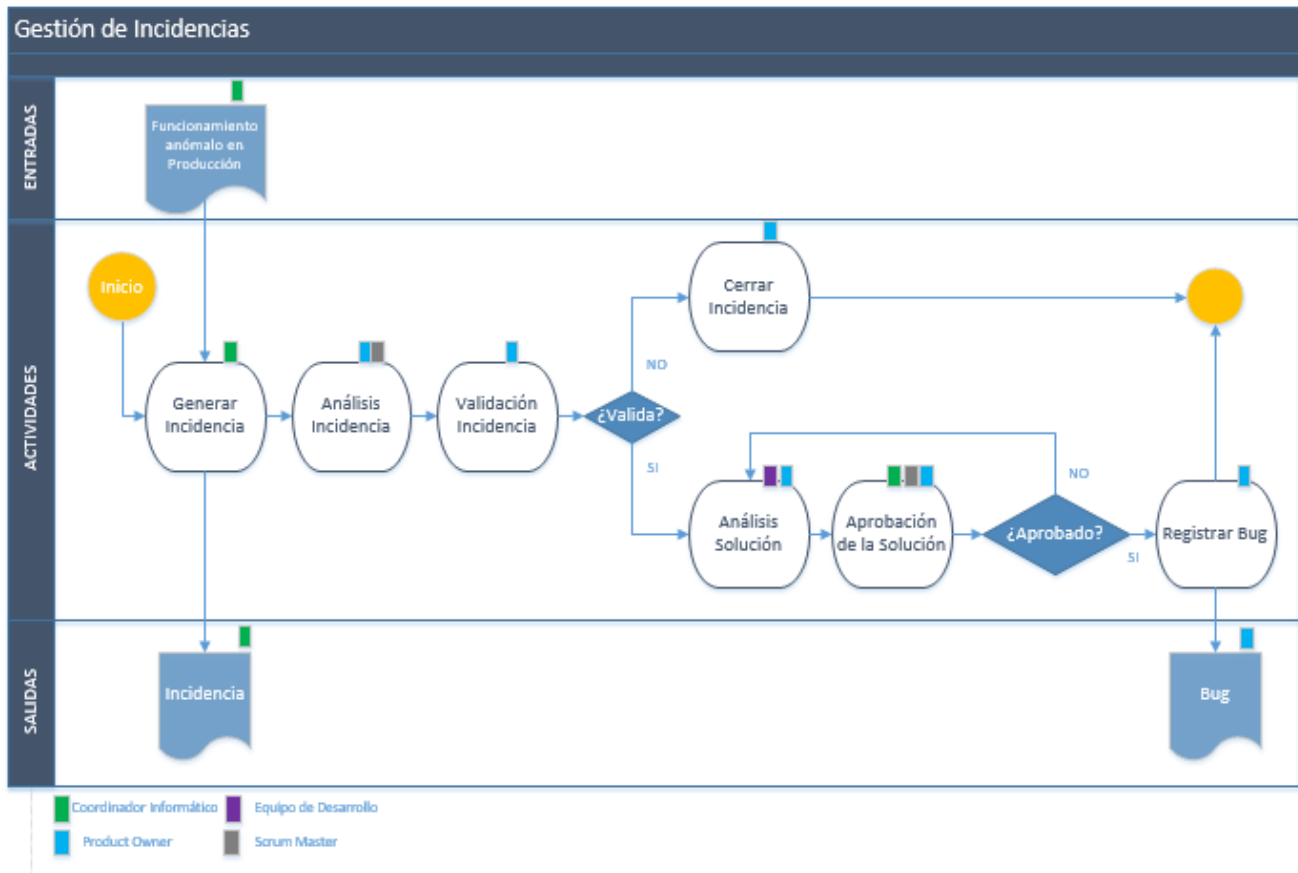


Figura 18: Gestión de Incidencias

Responsable
La responsabilidad de esta fase directiva recae sobre el <i>Product Owner</i> del proyecto
Actores Involucrados
<ul style="list-style-type: none"> Coordinador Informático <i>Product Owner</i> <i>Scrum Master</i> <i>Equipo de Desarrollo</i>
Entradas
Detección de un comportamiento anómalo, con la descripción más amplia posible sobre el alcance de la misma y los pasos para poder reproducirla.
Actividades
1. Se alerta de un mal funcionamiento del aplicativo. Esta alerta puede venir del Coordinador

<p>Informático, aunque excepcionalmente puede venir informada por el <i>Product Owner</i> o de cualquier miembro del <i>Equipo de Desarrollo</i>, canalizados por el <i>Scrum Master</i>.</p> <ol style="list-style-type: none">Entre el <i>Product Owner</i> y el <i>Scrum Master</i> identifican si se trata de una incidencia o de un requisito que deba seguir otro flujo:<ol style="list-style-type: none">Si resulta ser un requisito, finalizamos este proceso y se inicia un proceso de Gestión del Cambio.En caso de resultar una incidencia se continúa el proceso de Gestión de Incidencias.El <i>Equipo de Desarrollo</i> analiza la incidencia para aportar una solución a la misma.Se reúnen el Coordinador Informático, el <i>Product Owner</i> y el <i>Scrum Master</i>, para decidir la aprobación de la solución propuesta.<ol style="list-style-type: none">En caso de no aprobarse la propuesta se explican los motivos, y se solicita al <i>Scrum Master</i>, que adapte la solución a las nuevas indicaciones aportadas.En caso de aprobarse la solución, el <i>Product Owner</i> es el encargado de registrar los Requisitos del Producto necesarios para llevar a cabo su corrección.El <i>Product Owner</i> es el encargado de realizar su seguimiento hasta la solución del problema.	
Salidas	
<p>En cada una de las iteraciones de esta fase debe quedar constancia de la incidencia registrada, la solución propuesta y la ejecución de la misma. Deben almacenarse todas en un único punto aunque cada una se trate como un registro.</p>	

Tabla 3: Gestión de Incidencias

ii. Definición del Alcance

Definir el alcance es el proceso que consiste en desarrollar una descripción detallada del proyecto y del producto. El beneficio clave de este proceso es que describe los límites del producto o resultado mediante la especificación de cuáles de los requisitos recopilados serán incluidos y cuáles excluidos del alcance del proyecto.

En cualquier punto del proyecto se pueden producir cambios que afecten al desarrollo y alcance del mismo, ya sea por una necesidad surgida por los gestores que van a utilizar el aplicativo, como por parte del proveedor o del propio equipo de desarrollo. En estos casos se trataría de Cambios de Alcance.

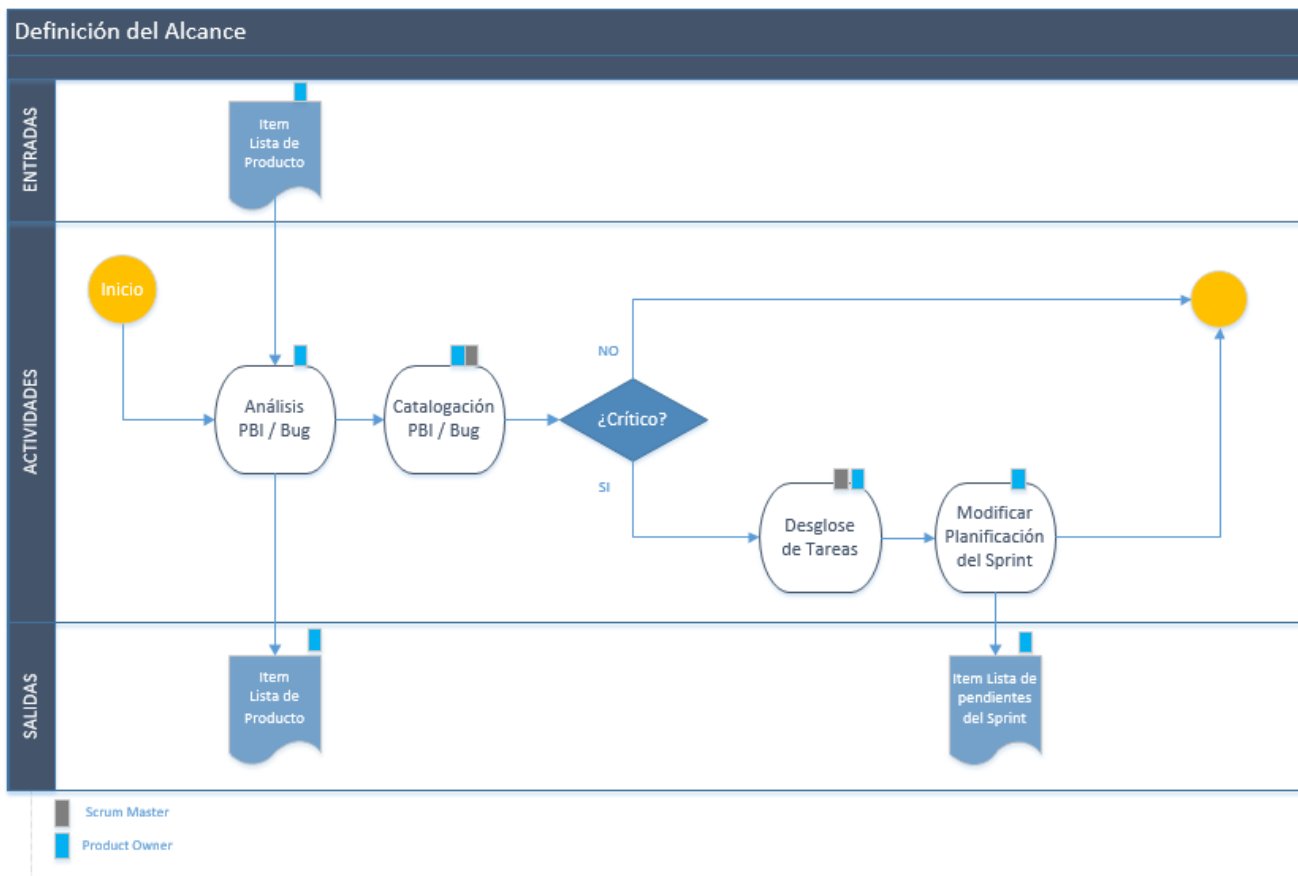


Figura 19: Definición del Alcance

Responsable
La responsabilidad de esta fase directiva recae sobre el <i>Product Owner</i> del proyecto
Actores Involucrados
<ul style="list-style-type: none"> <i>Product Owner</i> <i>Scrum Master</i>
Entradas
<ul style="list-style-type: none"> <i>Lista del Producto</i> que entre nuevo o se haya realizado algún cambio sobre su definición. Un <i>Bug</i>
Actividades
<ol style="list-style-type: none"> El <i>Product Owner</i> analiza la <i>Lista de Producto / Bug</i>, para calcular su impacto en el alcance del proyecto y aportar toda aquella información que aporte valor, implicando a los actores necesarios. El <i>Product Owner</i> analiza la prioridad y decide si fuera necesario o no incluirlo en el <i>Sprint</i> en curso, o bien esperar a nuevas planificaciones de <i>Sprint</i>. Durante esta tarea puede intervenir el Coordinador Informático para exponer sus objetivos y negociar la priorización de la <i>Lista de Producto</i>. Si se decide incluir, se debe desglosar en tareas e incluirlo en la revisión del alcance diario del <i>Sprint</i>.

Salidas

En cada una de las iteraciones de esta fase se debe actualizar la información de la Lista de Producto involucrada e iniciar aquellas revisiones de las que pudieran ser susceptibles a cambios.

Tabla 4: Definición del Alcance

c) Seguimiento

Durante todo el proyecto se realizarán reuniones de seguimiento periódicas con el fin de controlar la buena ejecución del mismo, y en el caso de necesitar cualquier actuación excepcional, detectarla con suficiente antelación para minimizar el impacto.

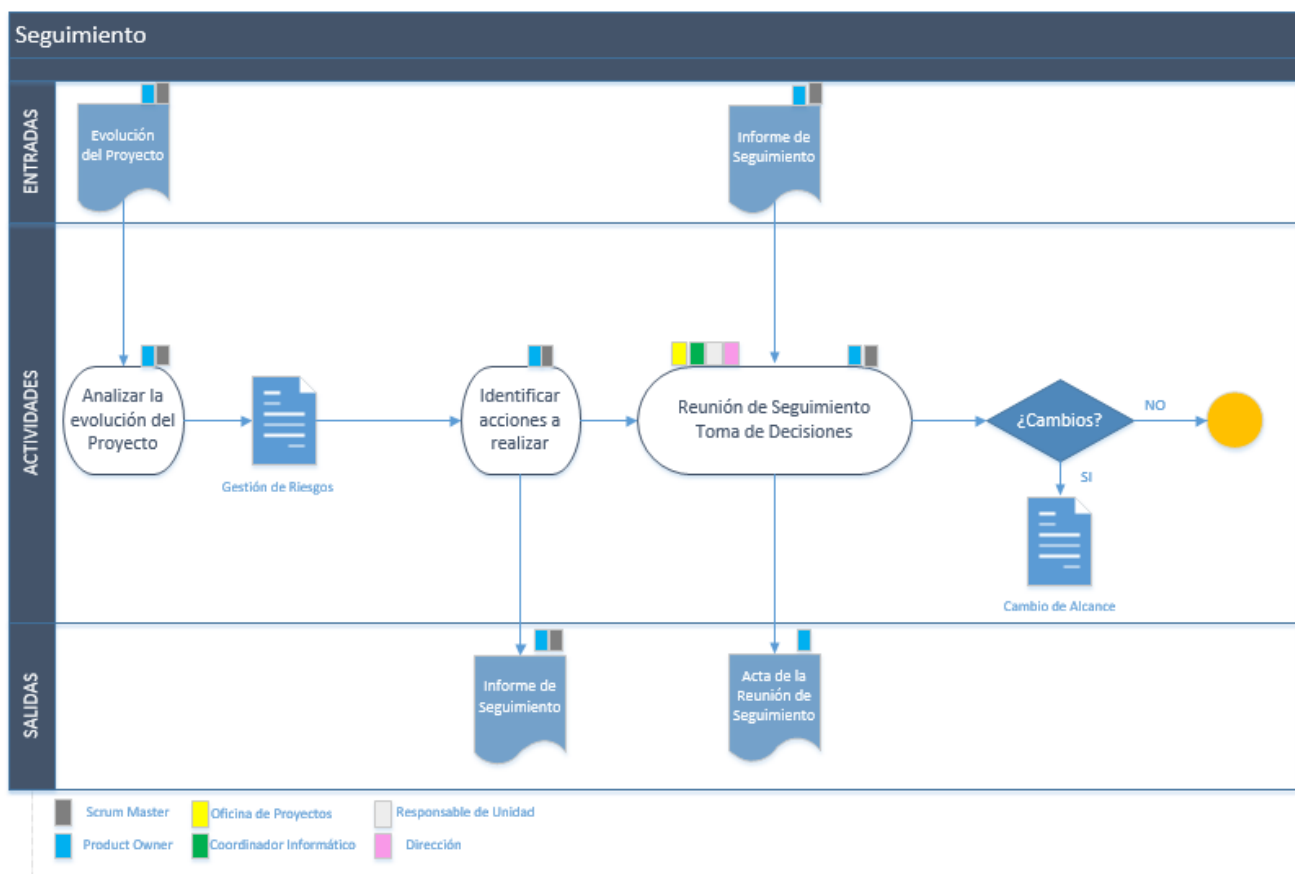


Figura 20: Seguimiento

Responsable

La responsabilidad recae sobre el *Product Owner*.

Actores Involucrados

- *Product Owner*

<ul style="list-style-type: none"> • <i>Scrum Master</i> • Coordinador Informático (opcional) • Oficina de Proyectos (opcional) • Responsable de la Unidad de Proyectos (opcional) • Director de Servicios (opcional)
Entradas
Informe de Seguimiento: Informe que plasma un fiel reflejo de la situación actual del proyecto para poder abordar los temas que requieran tomas de decisiones.
Actividades
<ol style="list-style-type: none"> 1. El <i>Product Owner</i> y el <i>Scrum Master</i> realizan un informe de seguimiento donde se plasma un análisis de la evolución del proyecto 2. Para completar dicho informe, se debe lanzar el proceso de Gestión de Riesgos para contemplarlos en el seguimiento 3. Una vez realizado el análisis de la evolución del proyecto y el estudio de los riesgos por los que se ve afectado, en caso de resultar necesario, el <i>Producto Owner</i> y el <i>Scrum Master</i>, identifican una serie de acciones que se deban acometer a partir de ese momento, a modo de propuesta para ser planteadas en la Reunión de Seguimiento. 4. Reunión de Seguimiento. Los convocados a esa reunión son el <i>Product Owner</i>, el <i>Scrum Master</i>, y de manera opcional en función de las necesidades, también el Responsable de la Unidad, el Coordinador Informático, el Director del Servicio y algún integrante del Equipo de Desarrollo, que por temas demasiado técnicos sea necesaria su presencia. En esta reunión se tomarán las decisiones oportunas en función de lo expuesto en el informe de seguimiento. <ol style="list-style-type: none"> a) En caso de tomar alguna decisión que provoque un cambio se iniciará el proceso de Gestión de Cambios b) De no ser así, se finaliza el proceso y no se realiza ninguna acción adicional
Salidas
Debe quedar constancia de todas las actuaciones realizadas durante este proceso, por lo tanto, como resultado final deben quedar el Informe de Seguimiento descrito en el proceso, y un acta de reunión que recoja todas las decisiones tomadas.

Tabla 5: Seguimiento

En la siguiente página se puede encontrar un ejemplo de un informe de seguimiento del proyecto:

CLIENTE	PROYECTO	FECHA
XXXX	Servicio de Orientación para la Búsqueda de Empleo y Creación de Puestos de Trabajo	
ASISTENTES	EMPRESA	

AGENDA

Punto de control de alto nivel de la evolución del proyecto.

NOTAS DE LA REUNIÓN Y PLAN DE ACCIÓN

Se ha tratado el estado actual del desarrollo del proyecto, el cual es satisfactorio actualmente para el CLIENTE.

A continuación se ha centrado la reunión en la identificación de los riesgos más señalados y que pudieran tener impacto sobre el proyecto:

1. Grafismo-Diseño gráfico: Se ha ido construyendo el estilo gráfico del portal de Búsqueda de Empleo, utilizando propuestas gráficas realizadas por la empresa Proveedora. Actualmente se da por cerrado el diseño gráfico actual, y se necesitarán pequeños retoques de diseño en las vistas sencilla y media y, posiblemente también en la ficha completa.

Acción: Se alargará la presencia del diseñador durante 5 jornadas.

2. Plataforma tecnológica del Portal de búsqueda: El CLIENTE está inmerso en el proceso de migración de sus plataformas a una versión de Windows más moderna que la actual, que es XP. Esto implica que las posibilidades de diseño gráfico, y experiencia de usuario de los navegadores modernos, soportados por esta versión del sistema operativo, permitirían funcionamientos del tipo *Responsive Design*, o plantillas para tabletas, móviles y similares.

NOTA: Se prevé que esta migración y despliegue se termine en dos meses, pero el Proveedor señala que la programación para Internet Explorer 8 es incompatible con la programación para plataformas modernas. Y que implicaría dos desarrollos diferenciados, con el sobre coste que pudiera implicar.

Acción: El Proveedor va a realizar tres estimaciones diferenciadas:

- Portal de búsqueda sobre Windows XP/IE8
- Portal de búsqueda sobre Windows 7 o superior, e IE10-Chrome
- Portal de búsqueda sobre tabletas de menos de 10 pulgadas, e IE10-Chrome-Safari

Figura 21: Ejemplo de Acta de Seguimiento

iii. Gestión de Riesgos

La gestión de Riesgos del Proyecto incluye los procesos para llevar a cabo la planificación de la gestión de riesgos, así como la identificación, análisis, planificación de respuestas. Para aumentar la probabilidad y el impacto de los eventos negativos del proyecto.

Es importante de manera periódica realizar una gestión de los posibles riesgos que afecten al proyecto, tanto de una manera directa como indirecta. Por ello, se debe realizar el proceso de gestión de riesgos previo a la gestión del alcance aprovechando los informes de control de alcance y el control de riesgos, obtenidos al final de cada iteración.

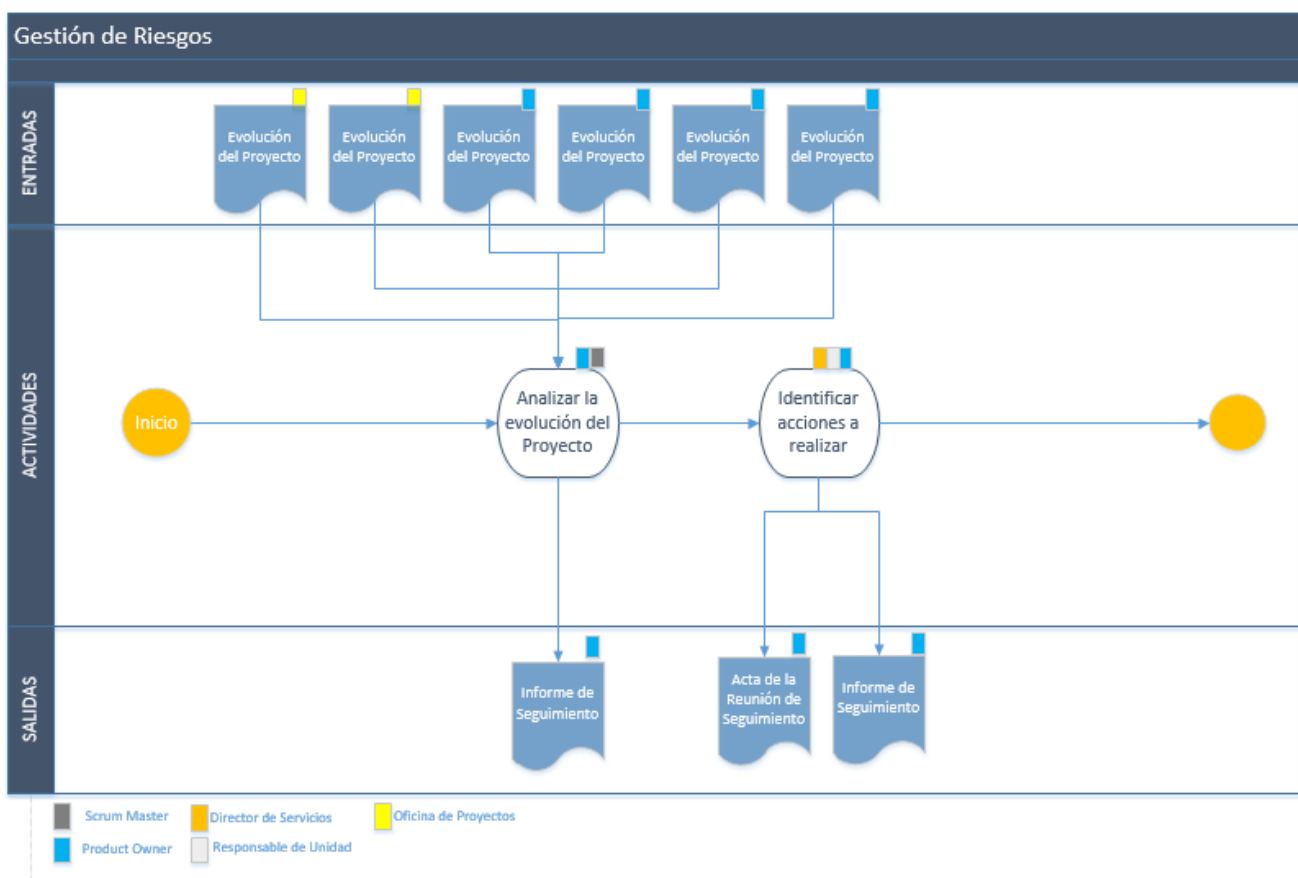


Figura 22: Gestión de Riesgos

Responsable

La responsabilidad recae sobre el *Product Owner*.

Actores Involucrados

- *Product Owner*
- *Scrum Master*
- Responsable de la Unidad de Proyectos

<ul style="list-style-type: none"> • Director de Servicios
Entradas
<p>La documentación necesaria para aportar una visión de contexto al proyecto es:</p> <ul style="list-style-type: none"> ○ Documento de Análisis de Necesidades, elaborado en el procedimiento de gestión de la demanda ○ Acta de Constitución del Proyecto ○ Lista de Producto ○ Catálogo de Requisitos ○ Informes de Control del Alcance ○ Catálogo de Riesgos
Actividades
<ol style="list-style-type: none"> 1. Tanto el <i>Product Owner</i>, por parte del cliente, como el <i>Scrum Master</i>, por parte del proveedor del Servicio, con la documentación del contexto del proyecto, identifican y clasifican los riesgos, atendiendo a los ámbitos (directos e indirectos) y a los diferentes impactos (alto, medio y bajo). 2. Una vez identificados, realizan un análisis enumerado de cada uno de ellos , con el fin de poder proponer acciones al respecto 3. Preparar una serie de propuestas de reacción a los riesgos, que en función de la gravedad de los riesgos, o de la probabilidad de que se produzca, pueden ser del siguiente tipo: <ol style="list-style-type: none"> a) Evasiva: Poder realizar alguna acción que nos permita evitar definitivamente el riesgo b) Mitigante: En caso de no poder evitar el riesgo, tratar de minimizar el impacto c) Transferida: Conseguir por medio de algún acuerdo o actuación que el riesgo sea asumido por terceros d) De aceptación: En ocasiones no es posible realizar ninguna de las acciones anteriores y hay que asumir el riesgo absorbiendo de la mejor manera posible el impacto e) En la planificación de respuesta a los riesgos se evalúa la necesidad de iniciar los planes de respuesta, para poderlos incluir en la gestión del alcance
Salidas
<p>El resultado final de esta fase a nivel documental es la sección del Informe de Seguimiento, y a nivel procedimental serán las medidas que se decidan adoptar.</p>

Tabla 6: Gestión de Riesgos

4.1.2.2. Fases Productivas

Se trata de las fases que forman el ciclo propiamente dicho. Se desarrollan de manera secuencial, en el orden que a continuación se establece:

d) Ingeniería de Requisitos

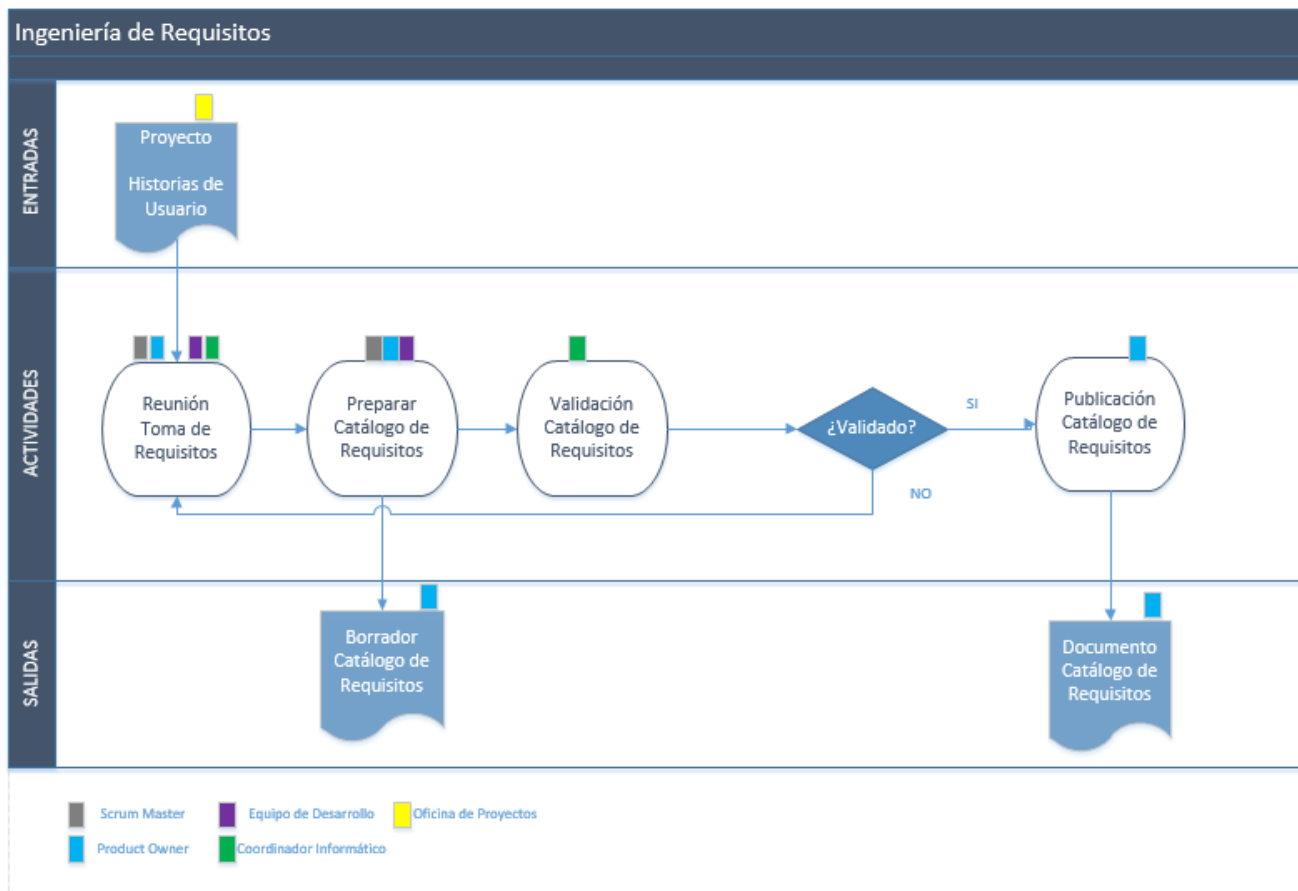


Figura 23: Ingeniería de Requisitos

Responsable
La responsabilidad recae sobre el <i>Product Owner</i> , que debe coordinar a todas las partes involucradas para obtener un Catálogo de Requisitos.
Actores Involucrados
<ul style="list-style-type: none"> • <i>Product Owner</i> • Coordinador Informático • <i>Scrum Master</i> (opcional) • Equipo de Desarrollo (opcional)
Entradas
Definición del proyecto con las correspondientes Historias de Usuario aprobadas por el comité de proyecto
Actividades
<ol style="list-style-type: none"> 1. El <i>Product Owner</i> convoca a una reunión de trabajo al Coordinador Informático, a quien él considere oportuno del Equipo de Desarrollo y opcionalmente al Responsable de la Oficina de Proyectos, para realizar una toma de requisitos a partir de las Historias de Usuario expuestas en el Anteproyecto aprobado. El Coordinador Informático debe aclarar todas las

dudas que surjan para que el Equipo de Desarrollo tenga la información suficiente para poder realizar una buena valoración de esfuerzo.

2. El *Product Owner* en colaboración con el *Scrum Master* y los analistas que considere necesarios, prepara un borrador del documento con el Catálogo de Requisitos, plasmando las descripciones de los requisitos y las pruebas de aceptación para cada uno de ellos. Es muy importante que cada uno de los requisitos cumplan los siguientes atributos:
 - a) Independiente del resto de requisitos en la medida de lo posible. Para poder seleccionarlo para un *Sprint* por el valor añadido que aporta al producto y no por estar forzado por una dependencia cuando por el valor que aporta aun no le correspondería.
 - b) Demostrable. De manera que sea claro cómo comprobar con el Coordinador Informático que el requisito está completado y que se cumplen sus expectativas.
 - c) De un grado de esfuerzo para ser completado semejante al del resto de requisitos. De manera que el Coordinador Informático pueda realizar una extrapolación del progreso del proyecto.
3. El Coordinador Informático revisa dicho borrador del Catálogo de Requisitos, produciéndose una de las siguientes situaciones:
 - a) El Coordinador Informático aprueba el Catálogo de Requisitos, quedando éste validado y pasando a la siguiente fase.
 - b) El Coordinador Informático plantea modificaciones debiendo volver al paso 2 para actualizar el Catálogo de Requisitos.

Salidas

Como resultado final de esta fase, debe quedar un documento de Catálogo de Requisitos creado por el *Product Owner* y validado por el Coordinador Informático. En este Catálogo se identifican, con un código único, cada uno de los requisitos solicitados para el desarrollo del proyecto, con el fin de poder realizar el seguimiento de manera efectiva.

Tabla 7: Ingeniería de Requisitos

e) nSprints

iv. Valoración y Planificación del *Sprint*

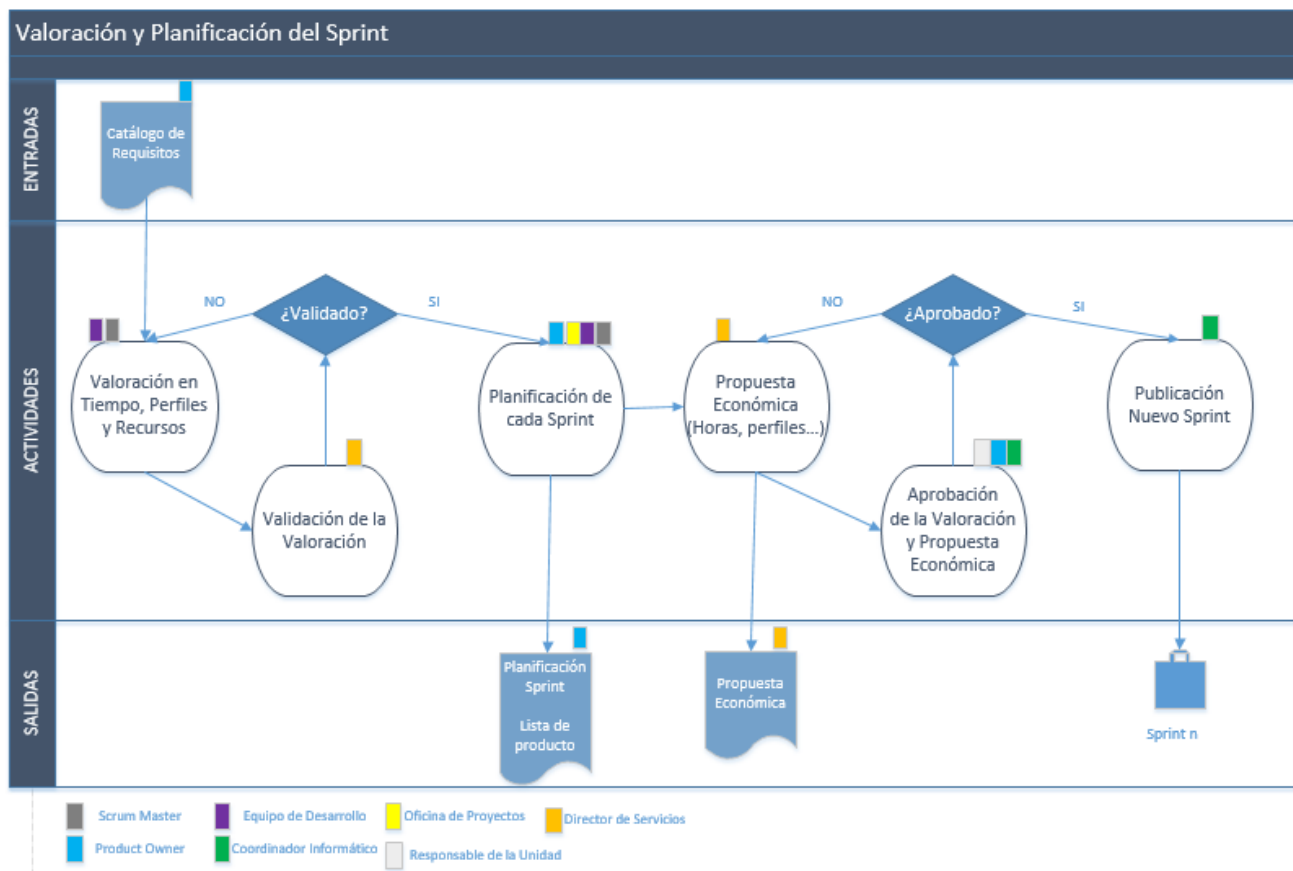


Figura 24: Valoración y Planificación del *Sprint*

Responsable
El <i>Product Owner</i> se encargará de esta fase, siendo el responsable de obtener una valoración en tiempo del Equipo de Desarrollo, una valoración económica del Director de Servicios y una Lista de Pendientes del <i>Sprint</i> como resultado de la planificación del <i>Sprint</i> .
Actores Involucrados
<ul style="list-style-type: none"> • <i>Product Owner</i> • Responsable de la Unidad de Proyectos Informáticos • Coordinador Informático • Director de Servicios • <i>Scrum Master</i> • Equipo de Desarrollo
Entradas
Catálogo de Requisitos con la suficiente descripción de cada uno de ellos como para poder realizar una valoración lo más ajustada posible a la realidad del caso.
Actividades

1. Entre el *Product Owner* y el *Scrum Master* se encargan de convertir el Catálogo de Requisitos en una serie de tareas que se conviertan en una Lista de Producto.
2. El Equipo de Desarrollo valora en tiempo cada una de las tareas de la Lista de Producto y es el *Scrum Master* el encargado de plasmarlo en un documento.
3. El *Scrum Master*, partiendo del documento de valoración generado, completa dicha valoración con la asignación primero de perfiles y finalmente de recursos para cada una de las tareas de la Lista de Producto. En este punto, implícitamente, ya se dispone de una valoración en coste, dividida por tareas, para poder tratar las prioridades con el Coordinador Informático.
4. Antes de mostrar la valoración al Coordinador Informático, ésta debe ser validada por el Director del Servicio. Se pueden producir dos casos, según la decisión del Director del Servicio:
 - a) Tiene discrepancias con la validación. Las plantea al *Scrum Master*, el cual actualiza la valoración en tiempo, en función de las conclusiones a las que lleguen.
 - b) Valida la valoración en tiempo. Continuamos al siguiente paso, Planificación de un nuevo *Sprint*.
5. Se reúne el Equipo de Desarrollo con el *Scrum Master* y opcionalmente se convoca al *Product Owner*, con el fin de realizar la planificación del *Sprint*, resultando la Lista de Pendientes del *Sprint*, con las correspondientes asignaciones de las tareas a cada miembro del Equipo de Desarrollo.
6. El Director de Servicios presenta una propuesta económica para el *Sprint* completo al cliente.
7. Reunión de trabajo entre el Coordinador Informático, *Product Owner* y Responsable de la Unidad, con el fin de aprobar la valoración en tiempo y propuesta económica presentada por el proveedor:
 - a) En caso de aprobación, se procese a ejecutar el *Sprint*, continuando con la fase de construcción.
 - b) En caso contrario se reúnen con el Director del Servicio para que aporte las aclaraciones pertinentes y poder llegar a un acuerdo.

Salidas

En esta fase se obtiene una Lista de Pendientes del *Sprint* perfectamente definida, con todas las tareas valoradas en tiempo, con una serie de prioridades establecidas, y una propuesta económica sobre el coste del *Sprint* completo.

Tabla 8: Valoración y Planificación del *Sprint*

En la página siguiente se puede observar un ejemplo de Lista de Producto.³

³ Se trata de una parte de la lista de producto del proyecto

CLIENTE	PROYECTO	FECHA
XXXX	Servicio de Orientación para la Búsqueda de Empleo y Creación de Puestos de Trabajo	

ITEM	DESCRIPCIÓN	ESTIMACIÓN
1	Preparación del entorno de trabajo	3
2	Registro y recopilación de datos del proyecto	3
3	Registrar y asignar nuevas tareas	4
4	Logueo de usuarios en el sistema	5
5	Formulario de Definición del proceso	3
6	Formulario de Medición y Análisis	3
77	Ejecución de Pruebas	5
78	Documentación	2
79	Formación	3
80	Aceptación y Garantía	3
81	Puesta en Producción	4
82	Generación entorno Pruebas	3
83	Inicio Garantía	2

Figura 25: Ejemplo de Lista de Producto

v. Desarrollo

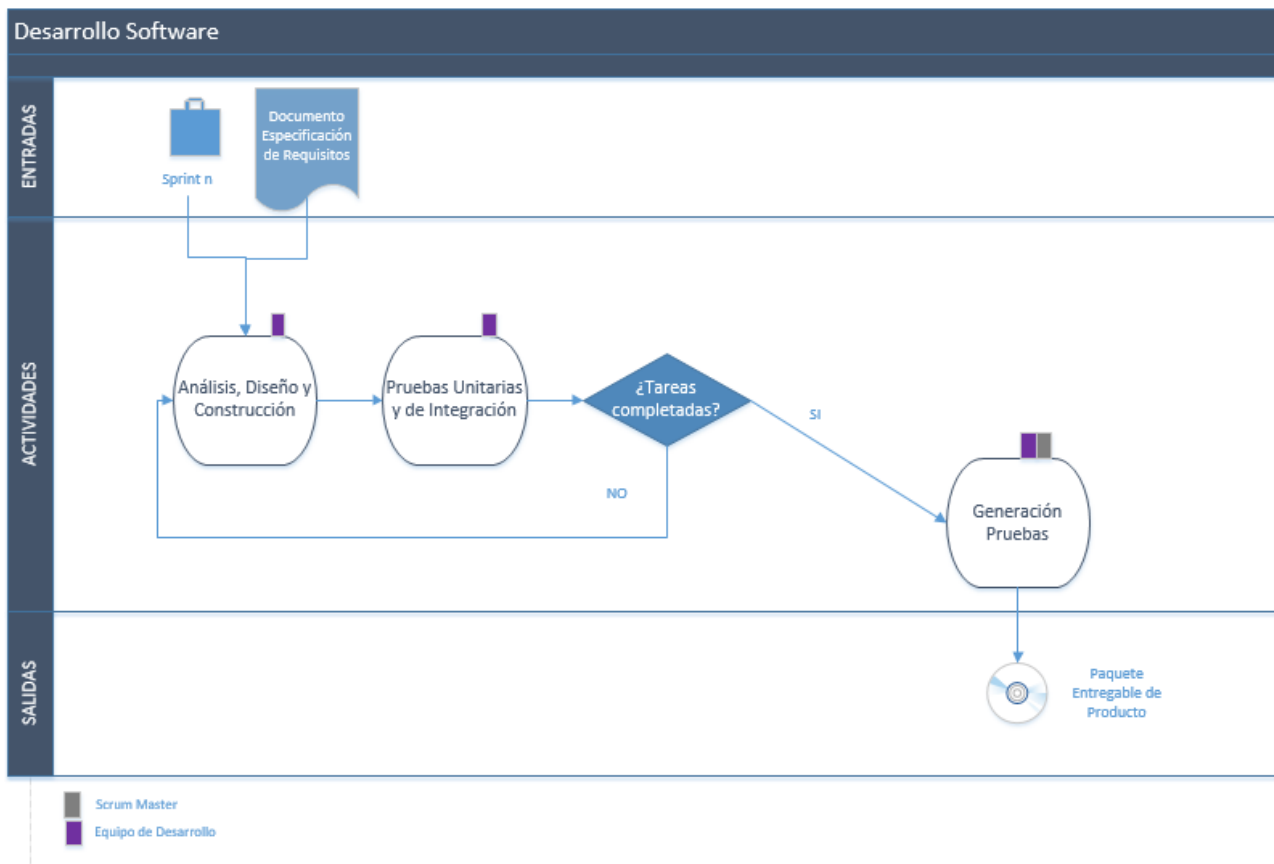


Figura 26: Desarrollo Software

Responsable
El <i>Scrum Master</i> se encargará de esta fase, siendo el responsable de que el <i>Sprint</i> que está en curso acabe de forma satisfactoria, completando todas las tareas incluidas en el mismo.
Actores Involucrados
<ul style="list-style-type: none"> <i>Scrum Master</i> Equipo de Desarrollo
Entradas
Lista de Pendientes del <i>Sprint</i>
Actividades
<p>Análisis, Diseño y Construcción</p> <p>Una vez que se haya definido y aprobado el documento de especificación de requisitos y se haya generado la planificación correspondiente a las tareas que se ejecutarán en el <i>Sprint</i> actual, el Equipo de Desarrollo comenzará con el análisis, diseño y construcción del software correspondiente a dichas tareas.</p> <p>Para que estas tareas den comienzo es necesario que los requisitos hayan sido previamente</p>

validados entre el *Product Owner* y el *Scrum Master*, para conocer el alcance real del *Sprint* en curso.

Durante esta fase se seguirán las directrices y buenas prácticas marcadas por el documento de arquitectura y se cumplirá con la normativa de Base de Datos.

Pruebas Unitarias

A medida que se lleva a cabo la construcción del software se irán ejecutando pruebas unitarias y de integración sobre el propio desarrollo para validar que su funcionamiento es correcto.

Las pruebas unitarias validan las funciones o métodos del código que se está construyendo, sin tener en cuenta el aspecto funcional de la tarea.

Las pruebas de integración validan la comunicación e interacción con otros componentes, módulos o sistemas.

Tras la ejecución de las pruebas unitarias y de integración se verifica si se ha completado el desarrollo de todas las tareas que estaban planificadas para el *Sprint* actual.

En caso afirmativo, se procederá a generar las pruebas correspondientes al desarrollo realizado, antes de llevar a cabo la subida al entorno de integración. De lo contrario se continuará con la construcción del software.

A pesar de ello la generación de pruebas es una tarea que puede comenzar durante el desarrollo, a medida que se van completando las tareas.

Salidas

El resultado final de esta fase consiste en un paquete parcial entregable del producto, que en función de la planificación del *Sprint*, debería funcionar independientemente de los desarrollos planificados en iteraciones posteriores.

Tabla 9: Desarrollo Software

vi. Despliegues

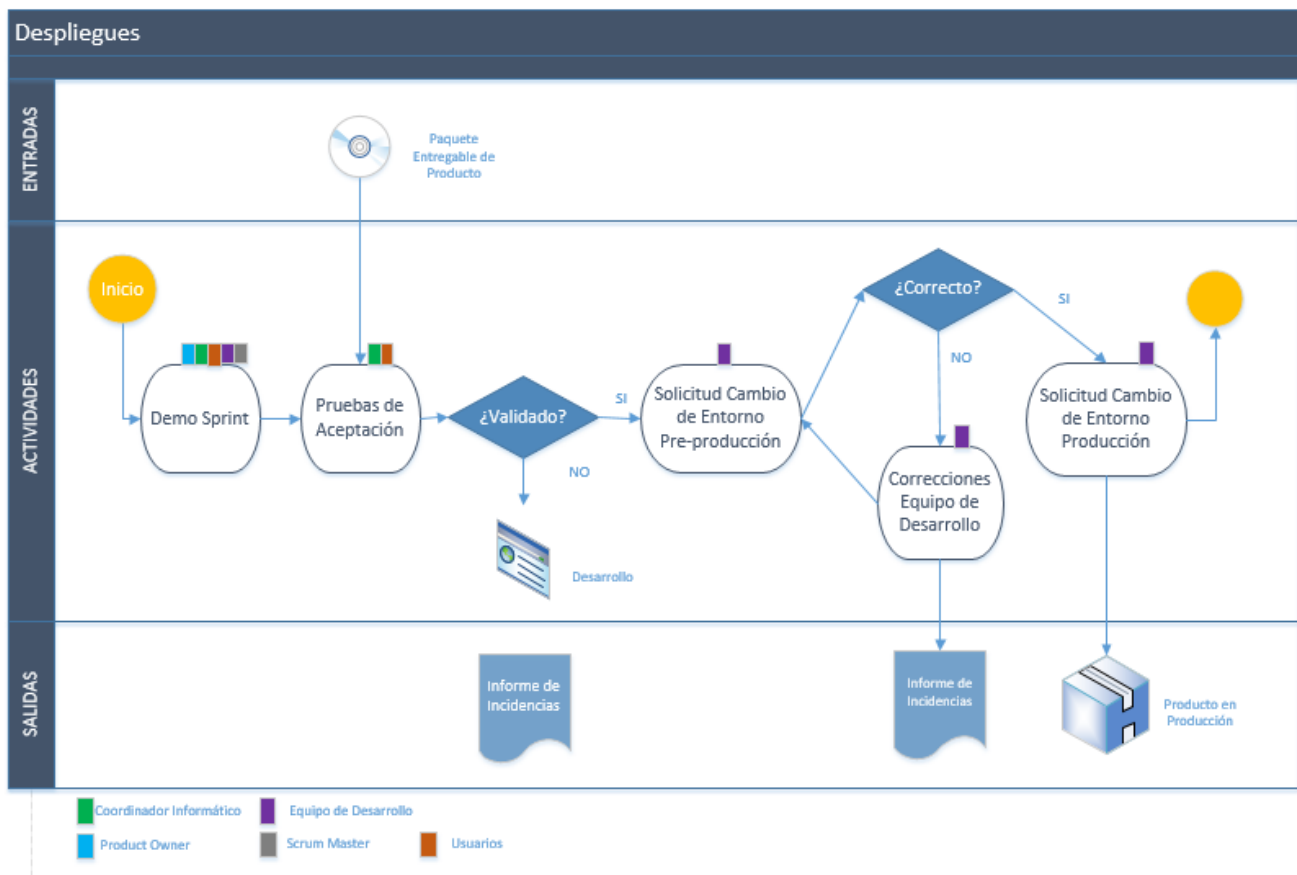


Figura 27: Despliegues

Responsable
Una vez generado un paquete entregable del producto, derivado de un <i>Sprint</i> , el <i>Scrum Master</i> es el encargado de realizar el seguimiento para que dicho entregable acabe desplegado en Producción.
Actores Involucrados
<ul style="list-style-type: none"> • <i>Product Owner</i> • Coordinador Informático • Usuarios • <i>Scrum Master</i> • Equipo de Desarrollo
Entradas
Una vez finalizado el <i>Sprint</i> debe realizarse un paquete entregable del producto para su posterior despliegue en los diferentes entornos del cliente. Este paquete del producto es precisamente la entrada que debe tener esta fase.
Actividades
1. Una vez que la aplicación ha sido verificada en el entorno de Integración por el Equipo de Desarrollo, se realizará la reunión de Revisión del <i>Sprint</i> actual. El <i>Product Owner</i> convoca al

<p>Coordinador Informático e invita opcionalmente a asistir a los Usuarios de la aplicación y a cualquier miembro del equipo de Desarrollo que considere necesario. En dicha revisión se comunica a los asistentes las tareas que han sido realizadas. A partir de ese momento el Usuario o Coordinador Informático podrá comenzar a ejecutar sus Pruebas de Aceptación en el entorno de integración.</p>	
<p>2. El Coordinador Informático o el Usuario realiza las Pruebas de Aceptación que se definieron en la Toma de Requisitos. En función del resultado de la ejecución de esas pruebas, puede ocurrir:</p> <ul style="list-style-type: none">a) Encuentra algún error o incidencia que debe resolverse antes de su puesta en producción. En este caso se retrocede a la fase de Desarrollo con el fin de solventar dichas incidencias.b) Todas las Pruebas de Aceptación resultan correctas, por lo tanto, valida el producto y da su visto bueno para que suba a producción. Continuamos al siguiente paso.	
<p>3. Una vez realizadas todas las verificaciones pendientes en el entorno de Integración, un miembro del Equipo de Desarrollo designado a ello, realiza la Solicitud de Cambio de Entorno, por la vía correspondiente, para que se despliegue en el entorno de pre-producción. Se debe entregar el paquete de producto con el que se realizó la Revisión del <i>Sprint</i>, con el código fuente y scripts de base de datos que sean necesarios.</p>	
<p>4. Cuando se haya realizado la subida al entorno de Pre-producción, el Equipo de Desarrollo con el soporte del Equipo de Explotación y Base de Datos, ejecutará las pruebas de arquitectura para verificar que las comunicaciones con los propios servicios, con otros sistemas, módulos, etc. funcionan correctamente.</p>	
<p>5. Una vez obtenido el visto bueno del Equipo de Desarrollo en el entorno de pre-producción, y siempre contando con la validación del <i>Product Owner</i>, un miembro del Equipo de Desarrollo designado para ello, realiza la solicitud de cambio de entorno a Producción por la vía correspondiente.</p>	
Salidas	
<p>El resultado final de esta fase es un nuevo producto funcionando en Producción, entendiendo por producto tanto un nuevo desarrollo como un evolutivo o correctivo de un desarrollo ya existente.</p>	

Tabla 10: Despliegues

4.1.3. Seguimiento y Control

A continuación se expone cómo realizar el seguimiento y control del Servicio durante cada uno de los ciclos de desarrollo.

Se propone que cada *Sprint* tenga una duración de tres semanas, con la posibilidad de modificarlo de una iteración a la siguiente en función de las necesidades de carga de trabajo u obligatoriedad de cumplimiento de algún plazo que provoque el acortamiento del mismo.

Se podrá cambiar la capacidad del equipo en base a los puntos de historia por *Sprint*. Cualquier cambio, salvo causa de fuerza mayor, sólo podrá realizarse entre iteraciones impidiendo que impacte en el *Sprint* en curso.

Se establecen las siguientes reuniones de seguimiento operativo:

- Reunión de Planificación del *Sprint*
- Reunión de Scrum Diario
- Reunión de Revisión del *Sprint*
- Reunión de Retrospectiva del *Sprint*
- Reunión de Certificación

4.1.4. Proceso de Certificación de los trabajos realizados

Se define el Proceso de Certificación de los trabajos realizados con el fin de poder facturarlos una vez al mes.

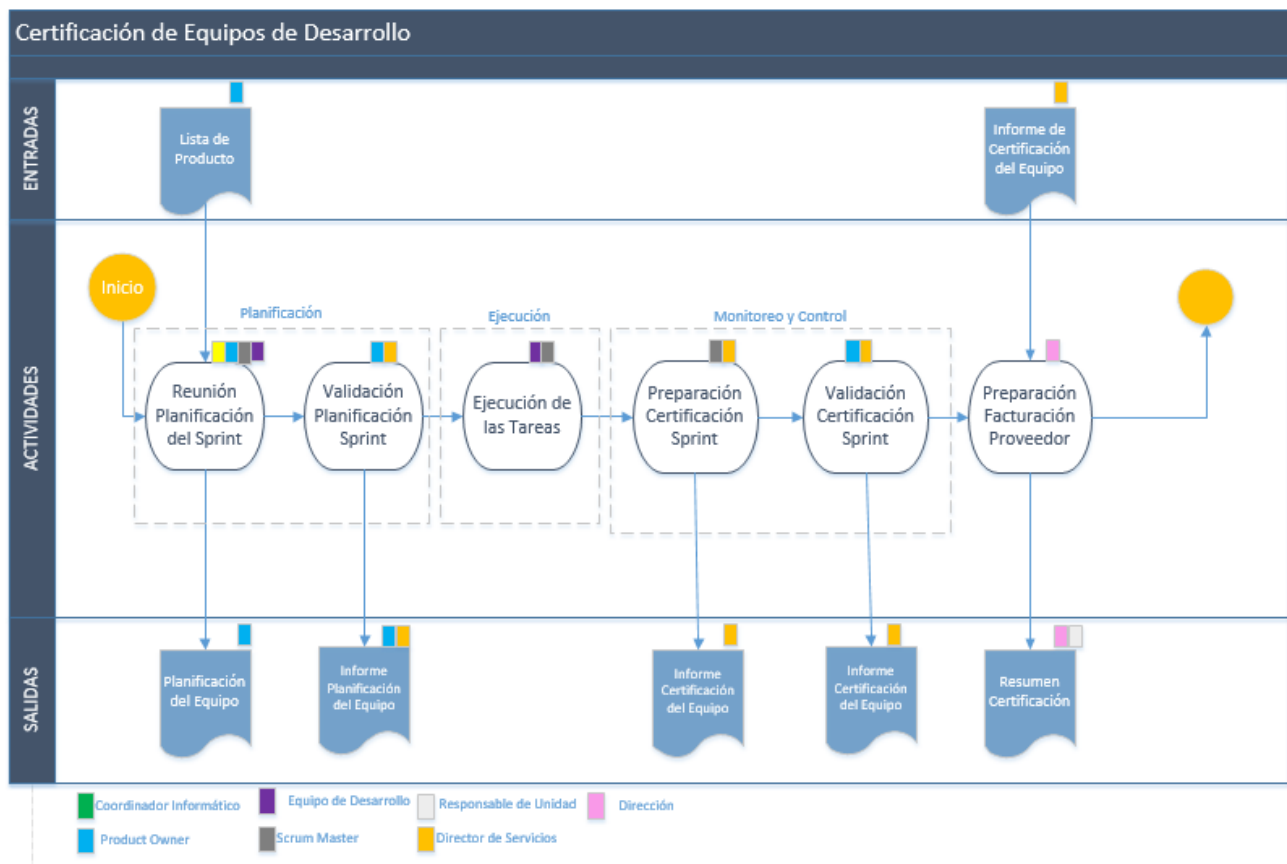


Figura 28: Certificación de Equipos de Desarrollo

Responsable
La responsabilidad de este proceso recae sobre el <i>Product Owner</i> del proyecto por parte del Cliente y en el Director del Servicio por parte del Proveedor.
Actores Involucrados
<ul style="list-style-type: none"> • <i>Product Owner</i> • <i>Scrum Master</i> • Equipo de Desarrollo • Responsable de la Unidad de Proyectos Informáticos • Apoyo Administrativo • Director de Servicio
Entradas
Lista de Producto
Actividades
<ol style="list-style-type: none"> 1. Durante los procesos de Planificación se realiza la Reunión de Planificación del <i>Sprint</i>, en la cual, partiendo de la Lista de Producto, se decide, estiman y planifican las tareas que se compromete a realizar el Equipo de Desarrollo en la iteración que se inicia en ese momento. Como resultado se obtiene un documento (Planificación del Equipo) que recoge todas las tareas comprometidas a finalizar en el plazo establecido para la iteración. 2. Tanto el <i>Product Owner</i> como el Director de Servicio validan el compromiso de las tareas planificadas y tras consensuarlas se genera el Informe de Planificación del cual se obtiene un documento PDF que recoge toda la información y que finalmente ambos aceptan firmándolo. 3. Comienza el proceso de Ejecución donde el Equipo de Desarrollo realiza las tareas planificadas a lo largo del <i>Sprint</i>, y puntualmente y estudiando cada caso, acepta nuevas tareas no planificadas, según las necesidades y la disponibilidad. 4. Una vez finalizado el <i>Sprint</i>, dentro de los procesos de Seguimiento y Control, el Director de Servicio con la colaboración del <i>Scrum Master</i>, prepara el informe de Certificación donde deben aparecer los siguientes datos: <ol style="list-style-type: none"> a) Todos los Requisitos de desarrollo a realizar para una aplicación. b) Todas las tareas tanto planificadas como no planificadas. c) Listado de las tareas finalizadas. d) Listado de las tareas no finalizadas y justificación si procede. e) Horas realizadas por la Línea Base. f) Resultado de los Acuerdos a Nivel de Servicio (ANS). g) Resultado de las pruebas realizadas. h) Listado de Coberturas de las pruebas realizadas. i) Listado de observaciones en caso de ser necesaria cualquier modificación manual del documento. 5. El <i>Product Owner</i> y el Director de Servicio verifican dicho informe y lo generan en PDF para firmarlo ambos a modo de validación del mismo. 6. Al final de cada mes, Apoyo administrativo prepara un único documento con el resumen de todas las certificaciones firmadas por el proveedor con el fin de justificar la facturación de

los trabajos realizados. Dicho documento lo firma el Responsable de la Unidad de Proyectos Informáticos dando de esta manera el visto bueno para proceder a su facturación.
Salidas
<ul style="list-style-type: none"> • Planificación del Equipo (Firmada por el <i>Product Owner</i> y el Director del Servicio). • Informes de Certificación (Firmados por el Director de Servicios y el <i>Product Owner</i>). • Resumen de los Informes de Certificación (Firmado por el Responsable de la Unidad de Proyectos Informáticos).

Tabla 11: Certificación de Equipos de Desarrollo

4.2. Modelo Organizativo

A continuación se detalla el Modelo Organizativo empleado en el caso de estudio:



Figura 29: Modelo Organizativo de un Servicio Gestionado

4.2.1. Configuración Equipo de Trabajo

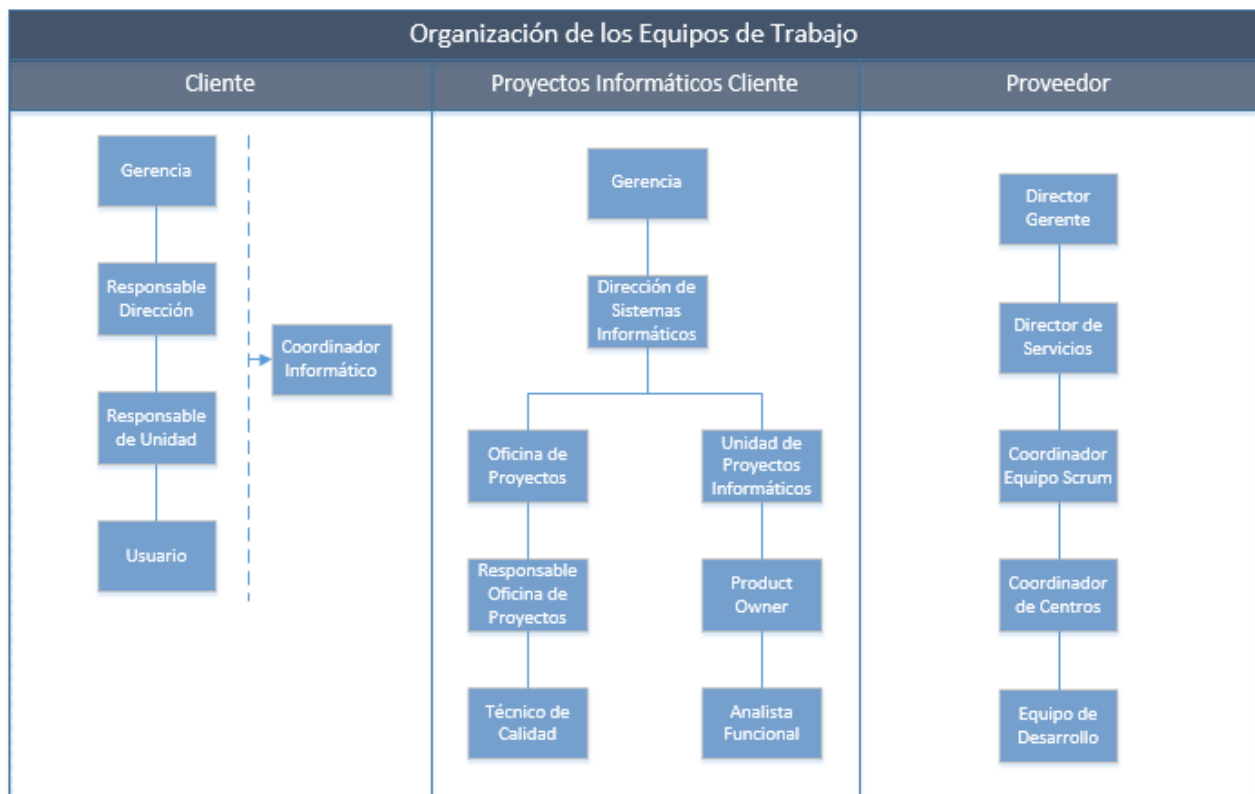


Figura 30: Organización de los Equipos de Trabajo

4.2.2. Roles y Responsabilidades

A continuación se exponen los roles que actúan en diferentes fases del proyecto durante el ciclo de vida de este, separados entre rol del Cliente y rol del Proveedor.

4.2.2.1. Roles del Cliente

- Dirección: Responsable de la Dirección de Sistemas de Información, de la que cuelgan las Unidades de Proyectos y Servicios Informáticos. Lidera el Comité de Dirección del Servicio y el Comité de Proyectos. Sus funciones son las siguientes:
 - Seguimiento del avance global del servicio.
 - Aprueba los cambios de los Acuerdos de Nivel de Servicio propuestos.

- Resolver los litigios que eventualmente pudieran producirse, y arbitrar conflictos o sucesos imprevistos que tengan un impacto en calidad, costes o retrasos en el proyecto.
 - Aprueba las modificaciones de la Línea Base.
 - Aprueba los cambios en el ámbito del servicio, que por su impacto o importancia así lo requiera.
- Responsable Unidad de Proyectos Informáticos (Responsable PI): Es el responsable máximo de los desarrollos que se realizan dentro de la Unidad, ya sea por desarrollo interno o por Servicio de Desarrollo Gestionado. Lidera el Comité de Seguimiento y forma parte del Comité de Proyectos. Sus principales funciones son:
 - Garantizar la consecución de objetivos y plazos.
 - Responsable de la coordinación, movilización e implantación de todos los participantes del Servicio.
 - Revisión de los elementos de Aseguramiento de Calidad, definición y seguimiento de los planes de mejora.
 - Aseguramiento del cumplimiento de los Acuerdos de Nivel de Servicio.
 - Toma de decisiones sobre la actuación a llevar a cabo de cada uno de los riesgos detectados.
- Responsable Unidad de Servicios Informáticos (Responsable SI): Es el responsable máximo de todos los servicios informáticos que la Dirección de Sistemas de Información proporciona al Cliente. Forma parte del Comité de Proyectos. Su equipo tiene las siguientes funciones:
 - Proporcionar soporte al equipo de desarrollo durante la definición y ejecución de las pruebas de arquitectura en el entorno de Pre-producción.
 - Gestión de las bases de datos y de la eficiencia en la utilización de las mismas.
 - Gestiona las solicitudes de cambio de entorno que solicita el equipo de desarrollo a los entornos de Pre-producción y Producción.
- Coordinador informático: Se trata de un gestor del Cliente que centraliza todas las solicitudes de los usuarios hacia la Unidad de Proyectos Informáticos que pueda derivar en un desarrollo, ya sea un nuevo proyecto, o un evolutivo o correctivo de un aplicativo ya existente.

Debe tener tanto conocimientos técnicos como de negocio para poder acometer las tareas asignadas a este rol. Entre otras, realizará las siguientes tareas o actividades:

 - Participa en la reunión de toma de requisitos, transmitiéndole las necesidades del usuario al *Product Owner* y otros miembros del Equipo de Desarrollo.
 - Interviene en la definición de las pruebas de aceptación del usuario, transmitiéndole al Equipo de Desarrollo los casos de uso de mayor importancia para el usuario.
 - Participa en la ejecución de las pruebas de aceptación del usuario en los entornos de Integración y Pre-producción.

- Cuando se detectan incidencias en el entorno de Producción envía la petición correspondiente al Equipo de Desarrollo con todo el detalle necesario para su tramitación y posterior resolución.
 - Valida formalmente los productos entregados en los entornos de Integración y Pre-producción.
- Product Owner: Es el responsable del resultado final del producto. Entendiendo como producto cualquier resultado medible, tangible y verificable que debe ser obtenido para completar un proyecto o parte de un proyecto. Algunas de sus tareas o actividades son:
 - Participa en la toma de requisitos tanto al principio de cada *Sprint* como durante la gestión del cambio, junto con el Coordinador Informático y aquellos miembros del Equipo de Desarrollo que considere oportunos, y realiza la publicación del Documento de Definición de Requisitos.
 - Participa en la definición de las pruebas de aceptación del usuario.
 - Gestiona la relación con el Usuario.
 - Dirige la reunión de Revisión del *Sprint*, presentándole al usuario las tareas que se han llevado a cabo.
 - Realiza la categorización de las incidencias reportadas y actualiza en la herramienta de gestión de peticiones las incidencias que han sido resueltas.
- Analista Funcional: Es el encargado de realizar el apoyo operativo del *Product Owner*. A continuación se exponen algunas de sus tareas o actividades:
 - Revisión de la definición de los requisitos establecidos, con el fin de detectar si son suficientemente claros, para poder realizar su análisis.
 - Revisión de las estimaciones realizadas por el Equipo de Desarrollo.
 - Revisión de los documentos de análisis entregados por el proveedor.
 - Ejecución del Plan de Calidad sobre los paquetes de producto entregados.
 - Control de Calidad de los entregables.
 - Colaboración en los trabajos del *Product Owner* que requieran una visión más técnica.
 - Realizar tareas de análisis funcional y técnico.
- Responsable de la Oficina de Proyectos (Responsable PMO): Es el coordinador de la Oficina de Proyectos. Sus principales funciones son:
 - Captación de los anteproyectos y presentación al Comité de Proyectos.
 - Proporcionar las herramientas y estándares a seguir en cada uno de los proyectos, tanto en la metodología Scrum como en la calidad.
 - Apoyo y seguimiento de los proyectos.

- Técnico de Calidad: Forma parte de la Oficina de Proyectos, y es el responsable de la parte de Calidad. A continuación se exponen las funciones, tareas o actividades que debe cumplir este rol:
 - Supervisa la correcta aplicación de las directrices marcadas por el Modelo de Calidad y la Estrategia de Pruebas durante todo el ciclo de vida de desarrollo del software.
 - Proporciona soporte al Equipo de Desarrollo durante la generación de las pruebas funcionales y de aceptación.
 - Proporciona soporte al Equipo de Desarrollo durante el diseño y mantenimiento de las pruebas automatizadas.
 - Supervisa la ejecución de las pruebas funcionales y de aceptación del Equipo de Desarrollo en el entorno de integración.

4.2.2.2. Roles del Proveedor

- Director Gerente: Es el responsable a Nivel Estratégico del proyecto por parte del proveedor del servicio. Sus principales funciones consistirán en:
 - Seguimiento del avance global del servicio.
 - Supervisión de los cambios de Alcance propuestos para el proyecto.
 - Consensuar con el Cliente los posibles cambios de los Acuerdos de Nivel de Servicio.
 - Aportar las necesidades oportunas para las eventuales modificaciones de la Línea Base.
 - Negociar los cambios en el ámbito del servicio, que por su impacto o importancia así lo requiera.
- Director de Servicios: Es el responsable de la empresa proveedora del Servicio de Desarrollo Gestionado. Sus funciones son:
 - Punto de unión, para la coordinación, entre el Cliente y el Proveedor.
 - Realiza la reunión de Certificación con el *Product Owner*.
 - Responsable de las valoraciones económica y la preparación para la facturación.
 - Debe dimensionar el Equipo de Desarrollo acorde con las necesidades acordadas en cada uno de los *Sprint*, según su planificación.
 - Responsable de la gestión del personal que conforma la Línea Base.
- Scrum Master: Es el responsable del Equipo de Desarrollo. Coordina las tareas de este equipo con el *Product Owner*. En caso que el equipo esté distribuido, el *Scrum Master* será el Coordinador responsable de la comunicación entre el *Product Owner* y el Equipo de Desarrollo. Para realizar esa labor, su trabajo se desempeñará desde las oficinas del Cliente. Sus principales tareas son:

- Dirige la fase de análisis, diseño y construcción del software.
 - Participa en la definición de las pruebas funcionales junto con los demás Analistas del Equipo de Desarrollo.
 - Dirige el diseño y mantenimiento de las pruebas automatizadas con la supervisión del Técnico de Calidad.
- Equipo de Desarrollo: Lo componen todas las personas que intervienen en la construcción del producto. Sus tareas, funciones o actividades son las siguientes:
 - Participan en las tareas de análisis, diseño y construcción del software.
 - Los Analistas del equipo definen las pruebas funcionales junto con el *Scrum Master* y con la supervisión del Técnico de Calidad, realizan la asociación de las mismas con los requisitos y generan el Documento de Definición de Pruebas.
 - Realizan los cambios de entorno en Integración, y las solicitudes de cambio de entorno en Pre-producción y Producción.
 - Realizan el diseño y mantenimiento de las pruebas automatizadas.
 - Ejecutan las pruebas funcionales y de usuario en el entorno de Integración, y las pruebas de arquitectura en el entorno de Pre-producción con el soporte del equipo de Explotación y Base de Datos.
 - Resuelven los errores detectados durante las pruebas realizadas en los entornos de Integración y Pre-producción.
 - Resuelven las incidencias reportadas por el Usuario en el entorno de Producción.

4.3. Modelo de Gestión del Servicio



Figura 31: Modelo de Gestión de un Servicio Gestionado

4.3.1. Modelo de Relación



Figura 32: Modelo de Relación entre Comités

4.3.2. Comités de Gestión

Para cada uno de los Niveles de Organización del Servicio se asigna un comité de gestión con el fin de realizar el seguimiento y control del servicio durante todas las fases de vida del proyecto.

Los comités de gestión son los encargados de llevar a cabo las reuniones de trabajo que se definirán a continuación, para garantizar una correcta prestación del servicio.

Se clasificarán los Comités de Gestión por niveles organizativos:

- Nivel Estratégico → Comité de Dirección
- Nivel Táctico → Comité de Seguimiento
- Nivel Operativo → Comité Operativo

4.3.2.1. Comité de Dirección

El Comité de Dirección forma parte del Nivel Organizativo Estratégico y es el responsable de la correcta prestación de servicio de todos los proyectos en los que un proveedor esté involucrado.

Objetivo
<ul style="list-style-type: none"> Realizar un seguimiento de la prestación de servicio ofrecida por el proveedor en todos los proyectos en los que esté involucrado. Agilizar la toma de decisiones que afecten a la consecución de los objetivos en plazo, calidad y coste. Aprobar cambios en el alcance y operativa. Exaltación de liderazgo del servicio.
Participantes
<ul style="list-style-type: none"> Dirección de Sistemas de Información Director Gerente Responsable de la Unidad de proyectos Informáticos (opcional)
Ubicación y periodicidad
Se realizará en las oficinas del Cliente a instancias de la Dirección de Sistemas de Información, cada cuatro meses o cuando la situación lo requiera.
Herramientas y Evidencias
Informe de seguimiento a nivel ejecutivo de la situación de los servicios prestados en el Cliente, tanto presentes como futuros con el fin de adelantarse a las necesidades que puedan surgir y que no afecte a la calidad del servicio. Un apartado más de este informe son los resultados de las ejecuciones de los Acuerdos a Nivel de Servicio.
Funciones y Responsabilidades
<ul style="list-style-type: none"> Seguimiento del avance global del servicio. Aprobar los cambios de los Acuerdos de Nivel de Servicio que hayan sido propuestos en el Comité de Seguimiento. Órgano responsable de la toma de decisiones estratégicas frente a posibles incidencias o cambios críticos. Resolver los litigios que eventualmente pudieran producirse, y arbitrar conflictos o sucesos imprevistos que tengan un impacto en aspectos de calidad, costes o retrasos en el proyecto. Es el responsable último de aceptación del servicio. Aprobar las modificaciones de la Línea Base. Aprobar los cambios en el ámbito del servicio, que por su impacto o importancia así lo requiera.

Tabla 12: Comité de Dirección

4.3.2.2. Comité de Seguimiento

En el nivel Organizativo Táctico se establece el Comité de Seguimiento cuya responsabilidad ya no recae sobre el desarrollo del proyecto, que para ello está el Comité Operativo, sino para garantizar que realmente se cumple con el nivel de servicio acordado entre el Cliente y el Proveedor del servicio.

Objetivo
<ul style="list-style-type: none"> • Garantizar la consecución de objetivos y plazos. • Controlar y coordinar la realización de las tareas. • Aseguramiento de la Calidad.
Participantes
<ul style="list-style-type: none"> • Responsable de la Unidad de proyectos Informáticos • Director de Servicio • Dirección de Sistemas de Información (opcional) • <i>Product Owner</i> (opcional) • Cualquier técnico del proyecto si la situación lo requiere (opcional)
Ubicación y periodicidad
Se realizará en las oficinas del Cliente a instancias del Responsable de la Unidad de Proyectos Informáticos, mensualmente o cuando la situación lo requiera.
Herramientas y Evidencias
Informe de seguimiento que plasme la situación actual del proyecto y las tareas planificadas futuras. En los puntos donde haya que tomar decisiones se pueden anotar las diferentes opciones a valorar dentro del mismo informe.
Funciones y Responsabilidades
<ul style="list-style-type: none"> • Responsable de la coordinación, movilización de todos los participantes del Servicio. • Gestión de la Planificación, incluyendo la definición, revisión y control del cumplimiento de la planificación. • Revisión de los elementos de Aseguramiento de la Calidad, definición y seguimiento de los planes de mejora. • Evaluar los riesgos, validando las medidas correctoras necesarias adoptadas. • Analizar el grado de avance de las tareas. • Revisar el Acuerdo de Nivel de Servicio medido en cada periodo, tratar la desviación del mismo y acordar en su caso las bonificaciones o penalizaciones aplicables. • Validar la calidad de la documentación necesaria para la reunión, informes de seguimiento, etc. • Análisis de riesgos y propuestas a adoptar para cada uno de los casos.

Tabla 13: Comité de Seguimiento

4.3.2.3. Comité Operativo

El Comité Operativo es el encargado de las reuniones necesarias en el Nivel Organizativo Operativo para el normal desarrollo de los *Sprint*.

Las reuniones mínimas que son responsabilidad de este comité y que se pasará a describir, son:

- Reunión de Planificación del *Sprint*
- Reunión de Scrum Diario
- Reunión de Revisión del *Sprint*
- Reunión de Retrospectiva del *Sprint*
- Reunión de Certificación

a) Planificación del *Sprint*

Objetivo
Planificar y distribuir los objetivos y alcance del proyecto en iteraciones, de manera que los requisitos estén priorizados balanceando el beneficio que aportan al Usuario, su coste de desarrollo y los riesgos del proyecto. De esta manera, las primeras iteraciones del proyecto podrán acomodar los requisitos más importantes y mitigar los riesgos más altos.
Participantes
La preparación de la Lista de Producto corre a cargo del <i>Product Owner</i> y el <i>Scrum Master</i> . En caso de no quedar claro alguno de los elementos de la Lista de Producto o alguna de las prioridades, se involucra al Coordinador Informático para eliminar cualquier duda que pueda provocar una mala planificación. Asistirá el Equipo de Desarrollo con el fin de valorar las tareas y asignarse las mismas. Con carácter opcional puede asistir tanto el Responsable de la Oficina de Proyecto como el Técnico de Calidad, siempre que la situación actual del proyecto lo requiera.
Ubicación y periodicidad
Se realizará en las oficinas del Cliente dado que todos los participantes desempeñarán su trabajo desde éstas. Se convoca cada 3 semanas a la finalización del <i>Sprint</i> anterior.
Herramientas y Evidencias
Se parte de la Lista de Producto, que debe estar priorizada por el Coordinado Informático. Se crea un nuevo <i>Sprint</i> y se enlazan los elementos de la Lista de Producto que se han seleccionado para componer este nuevo <i>Sprint</i> . De esta manera, queda definida la Lista de Pendientes del <i>Sprint</i> .
Funciones y Responsabilidades
Identificación de los objetivos del proyecto y de los requisitos iniciales de alto nivel que permiten la consecución de estos objetivos.

Priorización de los requisitos en iteraciones y entregas considerando los siguientes criterios:

- El valor aportado por cada requisito para el Usuario. Deberá ser explícito quién es el actor o usuario beneficiario de cada requisito y qué valor le aporta.
- El esfuerzo necesario para desarrollar cada uno de los requisitos, de manera que en las primeras iteraciones se desarrollen los requisitos que proporcionen el máximo valor con el mínimo esfuerzo y que puedan encajar en la periodicidad de las iteraciones.
- Las dependencias inevitables entre requisitos.
- Minimizar los riesgos del proyecto respecto al desarrollo de los requisitos, disponibilidad y grado de implicación en los actores y beneficiarios implicados, interacción con otros equipos (proyectos en paralelo, compras de material e infraestructura,...), etc.
- Maximizar la cohesión del contenido de cada iteración, identificando los puntos de acoplamiento y las dependencias entre los diferentes incrementos de manera que sean mínimos para poder dar por realmente completados los requisitos desarrollados en cada una de las iteraciones.
- Calcular la duración de cada uno de los incrementos desarrollados de manera que puedan encajar en la periodicidad de las tareas.

En el plazo comprendido entre la finalización de un *Sprint* y la planificación del siguiente, el Coordinador Informático prueba la nueva versión recibida y traslada las modificaciones o correcciones que ha ido detectando. En paralelo, el *Product Owner* y el *Scrum Master*, están elaborando la Lista del siguiente *Sprint* con las historias de usuario pendientes de desarrollar.

La Lista de Producto puede haber sufrido modificaciones solicitadas por el Coordinador Informático. Esto no es un problema, ya que es éste el que debe priorizarlo, pero sí se debe tener en cuenta.

El comienzo del siguiente *Sprint*, debe producirse a continuación del actual. Para el Equipo de Desarrollo, no debe haber periodo de tiempo entre uno y otro. Las tareas necesarias entre *Sprints*, serán abordadas por el equipo de la Línea Base, una vez finalizado el *Sprint*. En cualquier caso, estas tareas no deberían durar más allá de una semana, una vez finalizado el *Sprint*.

Tabla 14: Planificación del *Sprint*

b) Scrum Diario

Objetivo

El equipo de Desarrollo estudiará las tareas a realizar y las prioridades de las mismas, asignándose cada uno de los desarrolladores a sí mismo, qué tarea va a estar realizando a lo largo del día.

Participantes

Están organizados por el *Scrum Master*, que es el encargado de modelar la reunión y gestionar la pizarra en la que se va a tener la referencia visual del estado del *Sprint*. A esta reunión diaria, puede asistir el *Product Owner* y ver cómo va la evolución de las tareas que se van a desarrollar pero no puede intervenir en la misma ya que asistirá en calidad de oyente. Con esto se consigue que el *Product Owner* esté informado, en todo momento, de la evolución del desarrollo y se impide que añada nuevas tareas no planificadas al *Sprint* en curso.

Ubicación y periodicidad

Diariamente se celebrará una reunión a primera hora de la jornada laboral, de diez a quince minutos máximos. En un principio el servicio se proporciona desde las oficinas del Cliente, por este motivo, es fácil que el *Product Owner* asista a las reuniones diarias. En un futuro, con la posibilidad de deslocalizar el Equipo de Desarrollo, esta práctica podrá seguir llevándose a cabo con la utilización de alguna herramienta de conexión remota multimedia a modo de videoconferencia.

Herramientas y Evidencias

- Pizarra de Tareas

De gran importancia para la evolución y desarrollo de la metodología, va a permitir que cualquier persona con acceso a la sala de desarrollo pueda tener una idea de la evolución de los diferentes proyectos abordados por la unidad. El *Product Owner* debe tener acceso fácil para poder echarle un vistazo a la pizarra Scrum.

- Gráfico *Burndown*

Esta sería la herramienta de Seguimiento y Control en relación al desarrollo de un *Sprint* concreto. Proporciona una visión detallada de la situación actual del *Sprint* e información relativa al trabajo realizado por el Equipo de Desarrollo.

Se ejecuta el gráfico periódicamente en el que se vea la evolución de la resolución de las tareas a lo largo del tiempo que dure el *Sprint*.

Este gráfico es de gran importancia, ya que va a permitir conocer el estado actual del desarrollo, prever posibles desviaciones y medir un factor muy importante para usar con efectividad una metodología ágil, que es la velocidad de desarrollo del equipo.

Conocer la velocidad de desarrollo del equipo va a permitir tener una buena base para estimar tiempos, ya que mide la cantidad de trabajo que puede hacer el Equipo de Desarrollo durante un *Sprint*, lo que de cara a organizar sucesivos *Sprint*, va a tener una gran relevancia.

Funciones y Responsabilidades

Los miembros del equipo seleccionan las tareas de la columna "Pendiente" e irán desplazándolas a "En curso" cuando se las asignen y a la de "Terminado" cuando las hayan concluido, pasando a coger otra nueva tarea de la columna de "Pendiente" y continuar así con el desarrollo.

Se concede un espacio disponible en la pizarra que se reserva para ir recogiendo las tareas no planificadas que pueden haber surgido como consecuencia del propio desarrollo o como resultado de un requerimiento impuesto o de carácter urgente que, de común acuerdo con el Cliente, se decida incluir en el *Sprint* tras valorar que se puede abordar sin tirar abajo el mismo o acordando una demora que no debe exceder de una semana, para conseguir el objetivo del *Sprint* y realizar la entrega de valor.

El *Sprint* finaliza con una entrega de una nueva versión del producto, obteniéndose así una primera impresión con la que poder empezar a plantear el siguiente *Sprint*.

Tabla 15: Scrum Diario

c) Reunión de Revisión del Sprint

Objetivo
Mostrar el producto resultante del <i>Sprint</i> al Coordinador Informático y a todos los usuarios que se estime oportuno.
Participantes
El <i>Product Owner</i> lidera la demostración de las funcionalidades del producto, principalmente al Coordinador Informático. Deberían asistir todos los usuarios que vayan a utilizar el producto, pero si esto no fuera posible, deberán asistir los que el Coordinado Informático considere oportuno.
Ubicación y periodicidad
Se evitará el desplazamiento de los usuarios y el Coordinador Informático por lo que serán éstos los que propongan la ubicación de la revisión. Se celebrará cada finalización del <i>Sprint</i> . En principio cada tres semanas salvo que la situación del proyecto obligue a modificar el plazo de algún <i>Sprint</i> en concreto.
Herramientas y Evidencias
No hay mejor evidencia que el propio producto en funcionamiento. Para ello se despliega el producto en el entorno de integración, y se cargan los datos necesarios para poder mostrar los avances. Se descartan presentaciones o documentación en esta reunión, ya que lo importante es ver el funcionamiento real.
Funciones y Responsabilidades
Al finalizar cada iteración, el <i>Product Owner</i> realizará al Coordinador Informático y a los Usuarios que el Coordinado Informático considere, una demostración de los requisitos completados.
El Coordinado Informático hará una aceptación de estos requisitos realizando las comprobaciones de calidad oportunas, con los casos de prueba de Aceptación que se definieron en la primera iteración. Además el Coordinador Informático podrá volver a priorizar el conjunto de requisitos del proyecto y consensuará con el <i>Product Owner</i> el contenido de las siguientes

iteraciones según lo estipulado en la cláusula de Cambio de Alcance.

En particular, los elementos que se vuelven a priorizar serán: requisitos inicialmente identificados del proyecto, modificaciones a estos requisitos que aparezcan durante el proyecto, problemas de calidad detectados, etc.

Tabla 16: Reunión de Revisión del *Sprint*

d) Reunión de Retrospectiva del *Sprint*

Objetivo
Realizar una autoevaluación y autocrítica sobre todos los aspectos que han intervenido en el <i>Sprint</i> , tanto si ha sido positivo, para repetirlo en sucesivos <i>Sprint</i> , como si ha provocado un hecho negativo (retrasos, falta de calidad, errores...), para poder aprender de los errores y anticiparse a ellos en situaciones posteriores.
Participantes
Asistirán el <i>Product Owner</i> , el Equipo de Desarrollo y el <i>Scrum Master</i> .
Ubicación y periodicidad
En caso de ser posibles los desplazamientos, puede realizarse indistintamente tanto en las oficinas del Cliente como en las instalaciones donde esté el Equipo de Desarrollo trabajando.
Herramientas y Evidencias
<p>Durante la reunión se plantean una serie de preguntas sobre el <i>Sprint</i>, cuyas respuestas nos pueden indicar qué ha salido bien o mal y cómo podemos mejorarlo.</p> <p>Alguna de las cuestiones a las que hay que dar respuesta son las siguientes:</p> <ul style="list-style-type: none">• Si los miembros del equipo han trabajado bien juntos• Si el equipo hizo más o menos de lo previsto• Si el equipo tenía todas las habilidades y recursos que necesitaba para hacer el trabajo• Si los miembros del equipo entendieron los requerimientos• Si el equipo pudo completar el <i>Sprint</i> en línea con los requerimientos• Qué podría mejorarse o eliminarse en el siguiente incremento de funcionalidad• Qué se ha aprendido• Cuáles son los problemas que podrían impedirle progresar adecuadamente• Qué piensa el equipo de usar Scrum
Funciones y Responsabilidades
<p>1. Puesta en escena:</p> <ul style="list-style-type: none">○ El <i>Scrum Master</i>/ <i>Product Owner</i> explica el objetivo de la retrospectiva y enumera los

diferentes pasos y tiempos que va a tener la reunión.

2. Identificación de problemas:

- El *Scrum Master* hace un resumen de los hechos, objetivos conseguidos (o no) y decisiones más relevantes del periodo o proyecto a analizar con la finalidad de mejorar en el siguiente. Se apoya en datos objetivos utilizando para ello, por ejemplo, la lista de requerimientos priorizada, la lista de tareas de la iteración, las actividades realizadas, los gráficos de progreso del proyecto o de la iteración, etc.
- Los miembros del equipo proponen problemas adicionales mediante, por ejemplo, una lluvia de ideas (*Brainstorming*). El *Scrum Master* va anotando los problemas.
- El equipo agrupa los diferentes problemas según tengan una afinidad o relación y da nombre a cada grupo.
- El equipo vota qué grupos de problemas afectan más a los objetivos del proyecto.

3. Descubrimiento de causas:

- Para los grupos de problemas más votados, el equipo analiza por qué se están produciendo hasta encontrar las causas básicas.
- Notar que en ningún momento se trata de buscar culpables, si no de ver qué partes del proceso de trabajo, de interrelaciones y de colaboración, tanto en el equipo como con terceros, se deben mejorar para que el problema no se repita.

4. Plan de acción:

- El equipo propone soluciones para las causas que producen la mayoría de los problemas del proyecto. Para ello puede volver a utilizar la lluvia de ideas. Su selección puede estar basada en una votación o en criterios como coste, esfuerzo o tiempo de realización, satisfacción del cliente, sostenibilidad de la solución, etc.
- Se generan las tareas necesarias y se introducen en la lista de requisitos priorizada del proyecto (Lista de Producto) o la lista de tareas de la siguiente iteración (Lista de Pendientes del *Sprint*).

5. Qué ha funcionado bien

- Una retrospectiva no sólo debe fijarse en la solución de problemas, si no también hacer reconocimiento de las cosas que fueron bien e identificar las que hay que potenciar más. Para ello se puede aplicar un proceso muy similar al anterior.

6. Conclusiones de la reunión

- Se resumen las principales conclusiones y decisiones.
- Se revisa si se ha conseguido el objetivo de la reunión.
- Se analiza qué ha ido bien y qué hay que mejorar de la dinámica de la retrospectiva para

que la próxima sea mejor.

Tabla 17: Reunión de Retrospectiva del *Sprint*

e) Reunión de Certificación

Objetivo
El proveedor presenta al Responsable del contrato la justificación de los trabajos realizados a modo de certificación de los mismos.
Participantes
Como participantes obligatorios deben asistir el Director del Servicio y el <i>Product Owner</i> . A modo de supervisión y con carácter opcional, podrá asistir el Responsable de la Oficina de Proyectos. También se podrá convocar a cualquier persona tanto del proveedor como del Cliente que se considere necesario para la toma de decisiones en dicha reunión.
Ubicación y periodicidad
Se realizará en las oficinas del Cliente a instancia del <i>Product Owner</i> . Será cada cierre de <i>Sprint</i> y se certificará a parte del propio <i>Sprint</i> , la línea base consumida y los trabajos realizados.
Herramientas y Evidencias
Informe de seguimiento del cumplimiento de los trabajos durante el <i>Sprint</i> y justificación de la asistencia y desempeño de sus tareas correspondientes al proyecto del personal desplazado a las oficinas del Cliente.
Funciones y Responsabilidades
El director de Servicios presenta al Responsable del Contrato la justificación de los trabajos realizados durante el <i>Sprint</i> , tanto por parte del equipo de Desarrollo, como de los coordinadores o analistas del proveedor que desempeñan su labor en las oficinas del Cliente. Esta figura estará cubierta por el <i>Scrum Master</i> .

Tabla 18: Reunión de Certificación

4.4. Modelo Económico



Figura 33: Modelo Económico de un Servicio Gestionado

En función del tipo de Equipo de Trabajo involucrado en la realización de las tareas asignadas para llevar a cabo la ejecución del servicio, se definirán diferentes tipos de facturación.

A continuación se exponen los diferentes tipos de Equipos de Trabajo:

- **Coordinadores del Servicio:** Se trata de trabajadores del proveedor del servicio, que por motivos específicos de las tareas que van a realizar, es necesario que desempeñen su labor en las oficinas del Cliente, al menos en la mayoría de sus horas. Para este tipo de Equipo de Trabajo se define la Facturación Línea Base, que se explicará más adelante.
- **Equipo Scrum:** Se trata del equipo de desarrollo que se ocupa de todas las tareas detalladas en la Lista de Pendientes del *Sprint*, propias de cada uno de los *Sprint*. Los trabajos realizados por este equipo se repercutirán al Cliente por medio de la Facturación por *Sprint*.

4.4.1. Facturación Base

La facturación de la línea base dependerá de los perfiles de los trabajadores del Proveedor que realicen la mayoría de sus tareas en las oficinas del Cliente.

Estos trabajadores quedan asignados al cien por cien al proyecto que proporciona el servicio al Cliente. Independientemente si las tareas se realizan en el Cliente o de manera puntual en las oficinas del proveedor, se facturará por el número de horas asignado al proyecto durante el plazo establecido en la facturación de dicho periodo.

Se factura por mes natural. Al final de cada mes, el Director de Servicio presentaría un informe al *Product Owner* con el detalle de los trabajadores, sus perfiles y las horas en las que han prestado el servicio para su aprobación.

4.4.2. Facturación por *Sprint*

La facturación se realizará con la periodicidad con la que se establezcan los *Sprint*, sobre la base de los esfuerzos realizados por cada uno de los perfiles durante el periodo.

Como resultado de la reunión de planificación de cada *Sprint* se elaborará una Lista de Pendientes del *Sprint*, consistente en un documento con todas las tareas a realizar por el equipo de trabajo que proporciona el servicio durante el *Sprint*. Dicho documento se ampliará con la realización de una valoración de esfuerzos, plazo (correspondiente este último con la duración del *Sprint*), dimensionado del Equipo de Desarrollo y coste de los Puntos de Historia para ese *Sprint*.

El esfuerzo queda medido en Puntos de Historia para cada uno de los elementos de la Lista de Pendientes del *Sprint*. Los puntos de Historia son la unidad de medida de la facturación, que corresponde a un precio en función del dimensionamiento del Equipo Scrum, el Punto de Historia se define como el esfuerzo de una hora de trabajo de un único recurso perteneciente a los equipos Scrum.

El *Product Owner* revisará dichos informes de planificación prevista, que una vez validados servirán como primera versión de los anexos al informe de seguimiento del cumplimiento de los trabajos. Para realizar la facturación se realizará una Certificación de los trabajos realizados en el *Sprint*, dicha certificación recogerá la Lista de Pendientes del *Sprint* de la planificación y una Lista de Producto final donde aparecerán tanto las tareas planificadas como aquellas que hayan entrado de forma no planificada al *Sprint*.

5. PLANIFICACIÓN DEL PFC Y PRESUPUESTO

5.1. Planificación del PFC

A continuación se incluye una tabla con el detalle del desarrollo del proyecto, adicional a los apartados expuestos en el trabajo se han añadido las actividades previas al trabajo como son la elección del tema del proyecto y búsqueda de información del mismo para garantizar la viabilidad del proyecto antes de su ejecución.

Se puede observar de forma orientativa los días dedicados a cada una de las actividades, no se han tenido en cuenta de forma individual las revisiones y correcciones que se han hecho de los subapartados una vez finalizados sino que se ha tenido en cuenta una revisión total de toda la memoria que dura desde su inicio hasta el final.

La defensa del PFC será con fecha posterior al 5 de octubre de 2015.

Actividad	Fecha de Inicio	Duración	Fecha de Terminación
1. Actividades previas al comienzo del proyecto			
1.1. Búsqueda del tema y de información	01/06/2015	7	07/06/2015
1.2. Organización y estructura del documento	08/06/2015	2	09/06/2015
1.3. Reunión y exposición del tema a la tutora	19/06/2015	1	19/06/2015
2. Introducción			
2.1. Introducción	27/07/2015	2	28/07/2015
2.2. Objetivos	20/06/2015	1	20/06/2015
2.3. Estructura de la memoria	29/09/2015	1	29/09/2015
3. Metodologías Ágiles de Gestión de Proyectos			
3.1. Historia de los procesos de desarrollo	21/06/2015	5	25/06/2015
3.2. Metodologías Ágiles	26/06/2015	3	28/06/2015
3.3. ¿Cuándo se aplica una Metodología Ágil?	29/07/2015	1	29/07/2015
3.4. Manifiesto Ágil	29/06/2015	2	30/06/2015
4. Scrum			
4.1. Introducción	01/07/2015	3	03/07/2015
4.2. Roles de Scrum	04/07/2015	3	06/07/2015
4.3. Componentes	07/07/2015	3	09/07/2015
4.4. Eventos	10/07/2015	3	12/07/2015
4.5. Otras Herramientas	30/07/2015	3	01/08/2015
4.6. El proceso de Scrum	16/07/2015	4	19/07/2015
5. Caso de Estudio			

5.1. Modelo Operativo	03/08/2015	15	17/08/2015
5.2. Modelo Organizativo	30/08/2015	7	05/09/2015
5.3. Modelo de Gestión del Servicio	06/09/2015	9	14/09/2015
5.4. Modelo Económico	15/09/2015	3	17/09/2015
6. Conclusiones y Líneas Futuras			
6.1. Conclusiones	20/09/2015	2	21/09/2015
6.2. Líneas Futuras	22/09/2015	2	23/09/2015
7. Planificación del Proyecto y Presupuesto			
7.1. Planificación del Proyecto	27/09/2015	1	27/09/2015
7.2. Presupuesto	28/09/2015	1	28/09/2015
8. Revisión bibliográfica	01/06/2015	109	17/09/2015
9. Revisión de la memoria y entrega de la misma (no incluye la fecha de la defensa que será posterior a la entrega del PFC)	19/6/2015	109	05/10/2015

Tabla 19: Planificación del proyecto

La siguiente página contiene el Diagrama de Gantt con la planificación del proyecto según la tabla anterior.

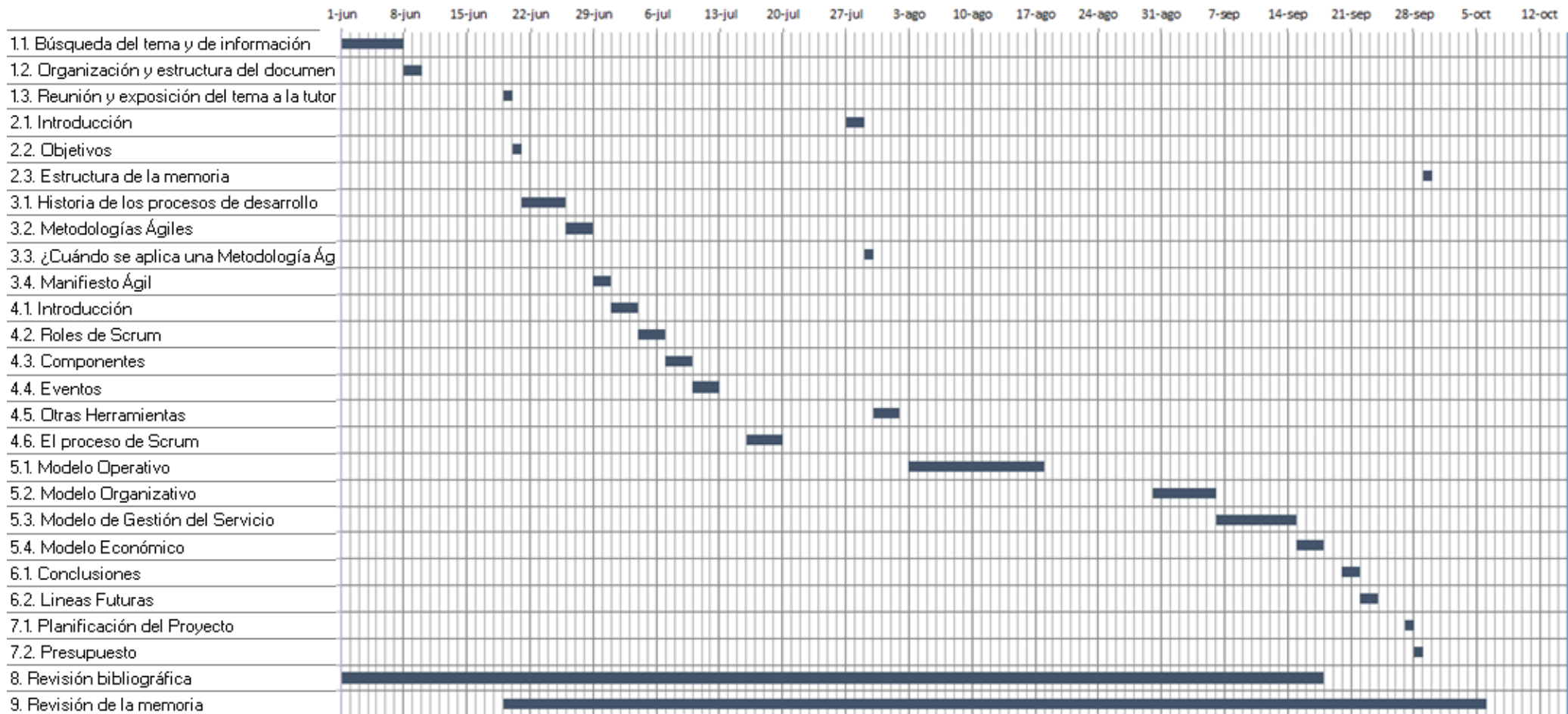


Figura 34: Gantt del proyecto

5.2. Presupuesto

El presupuesto global de la realización de este trabajo tiene en cuenta tanto los costes personales como los costes de los equipos y herramientas utilizadas para su desarrollo durante los cinco meses que ha durado su elaboración.

En las siguientes tablas se puede ver el detalle de los costes:

COSTES RECURSOS HUMANOS				
Nombre y apellidos	Categoría	Dedicación (personas mes)	Coste (€) (persona mes)	Coste total (€)
Raquel Pérez Leal	Ingeniera Senior	0,25	4.289,54 €	1.072,39 €
Ana Retamosa Santos	Ingeniera Junior	3	2.694,39 €	8.083,17 €
TOTAL				9.155,56 €

Tabla 20: Costes Recursos Humanos

COSTES EQUIPOS Y HERRAMIENTAS					
Descripción	Coste (€)	% Uso dedicado al proyecto	Dedicación (meses)	Periodo de depreciación	Coste Imputable (€) ⁴
Ordenador de sobremesa Compaq i3, 8GB RAM	499,00 €	100	5	60	41,58 €
Disco duro Western Digital 500GB	59,00 €	100	5	60	4,92 €
Licencias MS Office Professional	69,00 €	100	5	60	5,75 €
Licencia MS Visio Standard 2013	399,00 €	100	5	60	33,25 €
TOTAL					85,50 €

Tabla 21: Costes Equipos y Herramientas

⁴ Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

OTROS COSTES DIRECTOS	
Descripción	Coste total (€)
Papelería y suministro de oficina	30 €
TOTAL	30 €

Tabla 22: Otros costes directos asociados al proyecto

COSTES INDIRECTOS	
Descripción	Coste total (€)
Tasa de costes indirectos: 20%	1.854,21 €
TOTAL	1.854,21 €

Tabla 23: Costes indirectos asociados al proyecto

RESUMEN DE COSTES	
Descripción	Coste total (€)
Recursos Humanos	9.155,6 €
Amortización de Equipos y Recursos	85,5 €
Subcontratación de tareas	0,0 €
Otros Costes Directos	30,0 €
Costes Indirectos	1.854,2 €
TOTAL	11.125,30 €

Tabla 24: Resumen de Costes asociados al proyecto

El coste total del proyecto es de 11.125,30 €.

6. CONCLUSIONES Y LINEAS FUTURAS

6.1. Conclusiones

Tras realizar el estudio de las metodologías ágiles, en concreto Scrum, y visualizar un caso de estudio como el expuesto anteriormente, se enumeran y analizan las ventajas y desventajas que tienen las metodologías ágiles.

No existe una metodología universal para hacer frente con éxito a cualquier proyecto, toda metodología debe ser adaptada al contexto del propio proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, empresa que desarrolla el proyecto, etc.). Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de proyectos, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes. Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos que tienen estas características.

La siguiente tabla recoge esquemáticamente las principales diferencias de las metodologías ágiles con respecto a las tradicionales. Estas diferencias afectan no sólo al proceso en sí, sino también al contexto del equipo y a su organización.

METODOLOGÍAS ÁGILES	METODOLOGÍAS TRADICIONALES
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparadas para cambios durante el proyecto	Cierta resistencia a los cambios
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo o el contacto con éste es constante	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos de trabajo pequeños (<10 integrantes)	Grupos de trabajo grandes
Pocos artefactos. El modelo es prescindible	Más artefactos. El modelo es esencial
Pocos roles, más genéricos y flexibles	Más roles y específicos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	La arquitectura es esencial, se expresa mediante modelos y se define tempranamente en el proyecto

Tabla 25: Diferencias entre metodologías ágiles y tradicionales [14]

No se puede afirmar que una metodología es mejor que otra ya que cada proyecto implicará que se aplique una u otra. Analizando la metodología Scrum a lo largo de este trabajo, se puede garantizar que aportará los siguientes beneficios, siempre que se aplique en un proyecto apto para esta metodología:

- Gracias a su proceso iterativo, Scrum está **preparado ante los cambios de requisitos** que se produzcan a lo largo del desarrollo del proyecto. Es importante hacer bien la toma de requisitos pero igual de importante es poder modificar dichos requisitos sin ocasionar grandes pérdidas en cuanto a costes, motivación, tiempo, etc.
- **La modificación de los requisitos tienen un menor impacto** al realizar entregas parciales del producto final. Si el cliente quiere realizar cambios de una entrega respecto a la anterior, sólo se habrán perdido unas semanas de trabajo. En las metodologías tradicionales el cliente ve el producto una vez finalizado y los cambios sobre el mismo puede llevar a la pérdida de varios meses de trabajo.
- Las numerosas reuniones que el equipo realiza con el cliente hace que éste se involucre y comprometa a lo largo del proyecto, pudiendo observar cómo avanza el proyecto y aportar su experiencia para optimizar las características del producto final. De esta manera se **evitan numerosos malentendidos y aumenta la satisfacción del cliente** dado que poseerá en todo momento una completa visión del estado del producto.
- **La motivación e implicación del equipo de desarrollo es mayor** ya que todos los miembros del equipo conocen el estado del proyecto en cualquier momento, toda idea que surja de un integrante del equipo será tomada en cuenta y además los compromisos son negociados y aceptados por todos.
- Las metodologías ágiles en general permiten **ahorrar tanto tiempo como costes**, se trabaja de un modo más eficiente y rápido realizando entregas parciales pero funcionales del producto final.
- Es posible **mejorar la calidad del producto** al tener una continua interacción entre los desarrolladores y los clientes, de esta forma se puede asegurar que el producto final es lo que el cliente necesita y desea.
- Al realizar reuniones diarias, reuniones de revisión y reuniones de retrospectiva, se puede **detectar rápidamente errores y problemas**, y poner solución a los mismos de manera ágil y cooperativa entre los miembros del equipo. En las metodologías clásicas los errores no identificados en las primeras fases del proyecto suelen acarrear costes muy altos.
- Y, finalmente, las metodologías ágiles permiten **rentabilizar nuestras inversiones más rápidamente**. Gracias a la realización de entregas tempranas, el cliente tendrá rápido acceso a aquellas funcionalidades que en verdad aportan valor acelerando el retorno de la inversión.

Cuando se trabaja con las metodologías ágiles y con Scrum en concreto, se puede encontrar inconvenientes que hay que asumir:

- **Restricciones en cuanto al tamaño de los proyectos**, en general, es difícil aplicarlo en grandes proyectos. Está bien para proyectos pequeños, de rápido movimiento ya que trabaja bien sólo con equipos pequeños.
- **La falta de documentación del diseño** puede llegar a ser un inconveniente si el cliente exige una documentación exhaustiva o registro de los trabajos realizados. Hay proyectos bajo normativas internacionales que obligan a presentar documentación detallada del proyecto.
- **Problemas derivados de la comunicación oral**. Es muy fácil crear ambigüedad y provocar problemas entre el equipo y el cliente o incluso entre los propios miembros del equipo.
- **Fuerte dependencia de las personas**. Para que un proyecto triunfe siguiendo la metodología Scrum requiere que los miembros del equipo estén altamente formados y motivados. El equipo debe estar comprometido, ser capaz de tomar sus propias decisiones y autogestionarse. Además de los recursos sin suficiente experiencia, la falta de dirección firme pueden llevar a los proyectos a no completarse o incluso fallar.
- Al inicio de un proyecto de Scrum **no se puede predecir cuándo va a finalizar** ya que se facilita la realización de cambios en los requisitos lo que conlleva a variaciones en los tiempos. Realmente la aceptación de cambios debería verse como una ventaja tal y como se ha visto anteriormente, pero ¿Se prefiere un producto que se sabe con certeza que va a finalizar en 10 meses pero que está construido sobre las ideas y opiniones que se tenían cuando se comenzó? O quizás se prefiera un producto que podría acabar en el plazo similar pero que se ha podido evolucionar hacia nuevas necesidades reales y que se ha podido usar y probar antes de la entrega final.

6.2. Líneas Futuras

El camino que le queda a las metodologías ágiles por recorrer es amplio, y la posibilidad de trabajar varias metodologías ágiles juntas en un mismo proyecto o incluso combinar metodologías ágiles y predictivas en distintos sub-proyectos, hace que las posibilidades crezcan exponencialmente.

Cada vez son más los clientes que solicitan este tipo de metodologías para llevar a cabo sus proyectos y los proveedores que lo proponen como mejoras en sus propuestas. Como línea futura general para las metodologías ágiles y en concreto para Scrum, se propone la creación de un Estándar a través de la ISO (*International Standardization Organization*, entidad internacional encargada de favorecer normas de fabricación, comercio y comunicación en todo el mundo) que aplique a la organización y contemple tanto el proceso de Scrum como todos sus roles.

A día de hoy lo que se puede encontrar es la certificación del conocimiento de Scrum y gestión ágil como por ejemplo la certificación de Scrum Manager (primera certificación de Scrum, que se puso en marcha en 2007 [16]) o certificación de *Scrum Master*. Este es un servicio que ofrecen organizaciones privadas de carácter asociativo o empresarial como *Scrum Manager*, *Scrum.org*, *Scrum Alliance*, *ScrumStudy* entre otras. Las certificaciones emitidas por estas organizaciones responden a sus propios criterios de evaluación, y no son en ningún caso títulos oficiales de formación reglada.

Como línea futura particular para el caso de estudio expuesto en punto cuatro se propone realizar un Cuadro de Mando del *Sprint* como mejora.

Un cuadro de Mando es un repositorio centralizado de información referente al proyecto gestionado y que se alimenta en base a las métricas definidas en los Acuerdos de Nivel de Servicio. En él se recogen los objetivos de la iteración en curso, fechas de entrega y la velocidad a la que se está realizando el desarrollo, lo que permite la toma de decisiones sobre la estrategia del servicio y el cumplimiento de objetivos.

Del mismo modo, también permite evaluar tendencias, recogiendo métricas que abarquen diferentes niveles de seguimiento en función de las necesidades. Se puede consultar métricas diarias, por iteración, por periodos de facturación o aquellas que surjan de las reuniones de los diferentes comités de gestión del servicio.

7. BIBLIOGRAFIA

Para la realización de esta memoria se han tenido en cuenta las siguientes referencias marcadas a lo largo del texto:

- [1] http://scrummanager.net/files/gestion_proyectos_scrum_manager.pdf
(Último acceso septiembre 2015)
- [2] Leondes. "Intelligent systems: technology and applications", 2002.
- [3] Rubby Casallas, Andrés Yie. "Ingeniería de software: ciclos de vida y metodologías", artículo publicado en el sitio web de la Facultad de Ingeniería de la Universidad de Los Andes (Colombia).
- [4] S. Pressman, Roger. "Ingeniería del Software: Un enfoque práctico", 2001.
- [5] Boehm, Barry. "A Spiral Model of Software Development and Enhancement", 1986.
- [6] Boehm, Barry. "Anchoring the Software Process", 1995
- [7] Kent Beck. "Extreme Programming Explained: Embrace Change", 1999
- [8] <https://www.scrumalliance.org>
(Último acceso septiembre 2015)
- [9] <http://agilemanifesto.org>
(Último acceso septiembre 2015)
- [10] Hirotaka Takeuchi (Hitotsubashi University) y Ikujiro Nonaka. "The New Product Development Game", 1986.
- [11] Ken Schwaber y Jeff Sutherland. "Scrum Guide", 2013.
- [12] www.scrummanager.net/bok/index.php?title=Estimaci%C3%B3n_de_p%C3%B3quer
(Último acceso septiembre 2015)
- [13] <http://www.proyectosagiles.org/lista-tareas-iteracion-Sprint-backlog>
(Último acceso septiembre 2015)
- [14] Grupo ISSI. "Metodologías Ágiles en el Desarrollo de Software", 2003.
- [15] Project Management Institute. "Guía de los Fundamentos Para la Dirección de Proyectos (Guía del PMBOK®)", 2014.
- [16] <http://www.scrum-institute.org/>
(Último acceso septiembre 2015)

Además de las referencias citadas se han consultado los siguientes libros, artículos y páginas web:

Último acceso julio 2015:

- <http://ithgermanhdez.blogspot.com.es/2013/03/metodologias-de-desarrollo.html>
- <http://www.javiergarzas.com/2013/07/ciclos-de-vida-software.html>
- https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software
- https://es.wikipedia.org/wiki/Desarrollo_en_cascada
- <http://www.proyectosagiles.org/>
- <http://materias.fi.uba.ar/7500/schenone-tesisdegradoingenieriainformatica.pdf>
- https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software
- <http://www.northware.mx/wp-content/uploads/2013/04/Desarrollo-cascada-vs-Desarrollo-Agile.pdf>
- Jorge Fernández González. Introducción a las metodologías ágiles. Otras formas de analizar y desarrollar. (FUOC. Fundación para la Universitat Oberta de Catalunya)
- <http://agilemanifesto.org/iso/es/>
- <http://www.agilemanifesto.org/iso/es/principles.html>

Último acceso agosto 2015:

- http://ingenieriadesoftware.mex.tl/61154_ASD.html
- <https://es.wikipedia.org/wiki/Scrum>
- <http://www.scrummanager.net/>

Último acceso septiembre 2015:

- <http://www.marblestation.com/>
- <https://al095668.wordpress.com/2013/06/01/ventajasdesventajas-de-las-metodologias-agiles/> (Septiembre 2015)
- <http://comunidad.iebschool.com/iebs/agile-scrum/que-es-agile/>